RADC-TR- 71-30, Volume I
Final Technical Report
February 1971

HANDBOOK OF METHODS FOR INFORMATION SYSTEMS ANALYSTS AND DESIGNERS

Volume I - Basic Handbook and Appendix I

Synectics Corporation

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Synectics Corporation 4790 William Flynn Highway Allison Park, Pennsylvania 15101	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3. REPORT TITLE

HANDBOOK OF METHODS FOR INFORMATION SYSTEMS ANALYSTS AND DESIGNERS
Volume I - Basic Handbook and Appendix I

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5. AUTHOR(S) (First name, middle initial, last name)

James W. Altman Susan C. Shannon
Alvan W. Leavitt Stanford T. Hovey

6. REPORT DATE	7a. TOTAL NO OF PAGES	7b. NO OF REFS
February 1971	292	34

8a. CONTRACT OR GRANT NO F30602-70-C-0149 [redacted] Job Order No. 45940000 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 013-C-1 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-71-30, Volume I (of two)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Rome Air Development Center (INDA) Griffiss Air Force Base, New York 13440

13. ABSTRACT

A generalizable procedure for the analysis and design of information systems is described in the context of allied and supporting data methods, design assessment, and project management considerations. This procedure follows from a view of information systems development as a complex series of goal-directed iterations, rather than a well-ordered sequence of simple steps. In each iteration, tentative design alternatives are progressively narrowed, better defined, carefully assessed, and revised until a workable, user-responsive solution is operationally activated.

The analysis and design procedure is developed in two forms: (1) a comprehensive discussion of the basic concepts, rationale, and constructive operations supported by detailed flow diagrams; and (2) a simplified, convenient working tool (TRACE), illustrated with two sample system design problems of widely different complexity.

Handbook content and organization were evolved, uniquely, through provisions for systematic evaluation-refinement cycles at selected stages during the period of materials development. Potentially relevant materials were evaluated by a cross section of RADC research and development personnel with extensive practical experience in all facets of information systems development, who used techniques specifically adapted for this purpose. The resultant handbook constitutes a single-source, practice-oriented guide intended for those with formal training in the information sciences, but with little or no experience in military information systems development.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Systems Analysis						
Systems Design						
Information Systems						
Handbook						

HANDBOOK OF METHODS FOR INFORMATION SYSTEMS ANALYSTS AND DESIGNERS

Volume I - Basic Handbook and Appendix I

James W. Altman
Alvan W. Leavitt
Susan C. Shannon
Stanford T. Hovey

Synectics Corporation

FOREWORD

This final technical report in handbook form was prepared by Synectics Corporation (formerly Datagraphics, Incorporated), 4790 William Flynn Highway, Allison Park, Pennsylvania, under contract F30602-70-C-0149, Job Order Number 45940000, for Rome Air Development Center, Griffiss Air Force Base, New York. Mr. William Doig (INDA) was the RADC Project Engineer. Contractor's identification number is 013-C-1.

The authors are indebted to those many Rome Air Development Center personnel who participated with interest and vital effort in the unique process chosen for shaping this handbook, particularly Mr. Samuel DiCarlo, Mr. Doig, Miss Patricia Langendorf, and Mr. Roger Weber.

This report has been reviewed by the Information Office and is releasable to the National Technical Information Service.

This technical report has been reviewed and is approved.

Approved: William A. Doig
WILLIAM A. DOIG
Project Engineer
Engineering Analysis & Applications Section

Approved: F. H. Dettmer
FRANZ H. DETTMER
Colonel, USAF
Chief, Intel and Recon Division

FOR THE COMMANDER: Fred I. Diamond
for IRVING J. GABELMAN
Chief, Advanced Studies Group

ABSTRACT

A generalizable procedure for the analysis and design of information systems is described in the context of allied and supporting data methods, design assessment, and project management considerations. This procedure follows from a view of information systems development as a complex series of goal-directed iterations, rather than a well-ordered sequence of simple steps. In each iteration, tentative design alternatives are progressively narrowed, better defined, carefully assessed, and revised until a workable, user-responsive solution is operationally activated.

The analysis and design procedure is developed in two forms: (1) A comprehensive discussion of the basic concepts, rationale, and constructive operations supported by detailed flow diagrams; and (2) A simplified, convenient working tool (TRACE), illustrated with two sample system design problems of widely different complexity.

Handbook content and organization were evolved, uniquely, through provisions for systematic evaluation-refinement cycles at selected stages during the period of materials development. Potentially relevant materials were evaluated by a cross section of RADC research and development personnel with extensive practical experience in all facets of information systems development, who used techniques specifically adapted for this purpose. The resultant handbook constitutes a single-source, practice-oriented guide intended for those with formal training in the information sciences, but with little or no experience in military information systems development.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

SECTION I

ORIENTATION

The two chapters contained in this section describe the handbook content first in structural and then in conceptual terms. Chapter 1, The Handbook, examines factors related to the use of the handbook:

1. What is its purpose?
2. Who are its intended users?
3. What are some broad implications for users?
4. What is its content?
5. How is it organized?
6. How is the handbook most effectively used?

Chapter 2, Systems, Process, and Products, presents the conceptual framework of the handbook content in relation to system design and development. The emphasis of this chapter lies in three general areas:

1. To define and describe information system characteristics and design attributes, providing a conceptual goal for design and development activities.
2. To provide a design and development strategy for achieving that information system concept.
3. To identify the products (formal and informal) which result from that design and development strategy.

CHAPTER 1

THE HANDBOOK

This Handbook of Methods for Information Systems Analysts and Designers looks toward a class of design problems without peer in challenging the skills and creativity of systems development and implementation personnel. As an aid to meeting this challenge, the handbook presents a comprehensive and consistent set of procedures with which any particular information system requirement may be approached. Thus, at one extreme, the requirement may be a limited improvement to an operational capability already in being, while at the other, it may envision a wholly new system. The handbook offers a design approach, applicable to all possibilities, which is described at three specificity levels. That is, in terms of:

1. Basic concepts and relevant methodologies.
2. A generalized design-development strategy for proceeding from requirements definition to full operations.
3. Step-by-step examples of the procedures applied to widely different system requirements.

This handbook is intended, primarily, for use by two vitally important groups of participants in systems research, development, and implementation. The first includes those scientific-technical specialists, who are new or very recently assigned, to information system design activities. The second group comprises project level managers and supervisors of systems design activities who are new to, or unfamiliar with, the unique demands of complex information processing and handling problems. It was assumed in this connection that members of both groups would bring a background of formal training and some knowledge or experience in one or more areas related to the information sciences. Hence, the handbook often references, but does not detail, the nature and use of such "tools of the trade" as mathematical and statistical techniques, electronic design techniques, research design principles, and the like. It was also assumed that research and development policy directives and administrative practices were already available to these groups and need not be repeated.

Unquestionably, the handbook may be used in a wide variety of ways. Certain major applications were anticipated in its preparation and are cited below:

1. To orient and guide the practice of systems development participants.

2. To assist in planning, organizing, and managing systems development efforts.

3. To orient user agency representatives and others who interface with systems development efforts.

4. To establish a frame of reference for standardizing system design procedures, staff requirements, and team organization.

Handbook organization is shown schematically in Figure 1-1. This chart may be used to locate the principal content and subject areas treated in the text. The contents are structured in four sections with brief introductions which explain the scope and purpose of those chapters contained in the section. Where appropriate, the chapter introduction is accompanied by a more detailed aid to content location similar to Figure 1-1. In general, handbook organization can be understood by reference to Figure 1-2. Here, the core Chapters 6-10 (Section III) which consist of the generalized design procedures, are shown supported by the design effort implementation considerations in Chapters 3, 4, and 5 (Section II). Section IV provides further support in the form of two Appendices. Appendix I is a list of sources of design-relevant information. Appendix II is devoted to TRACE (Total Requirements Analysis for Concept and Elements) with two case studies illustrating its application. In essence, TRACE represents a practical method for carrying out the concepts and procedures presented in the main body of the handbook.

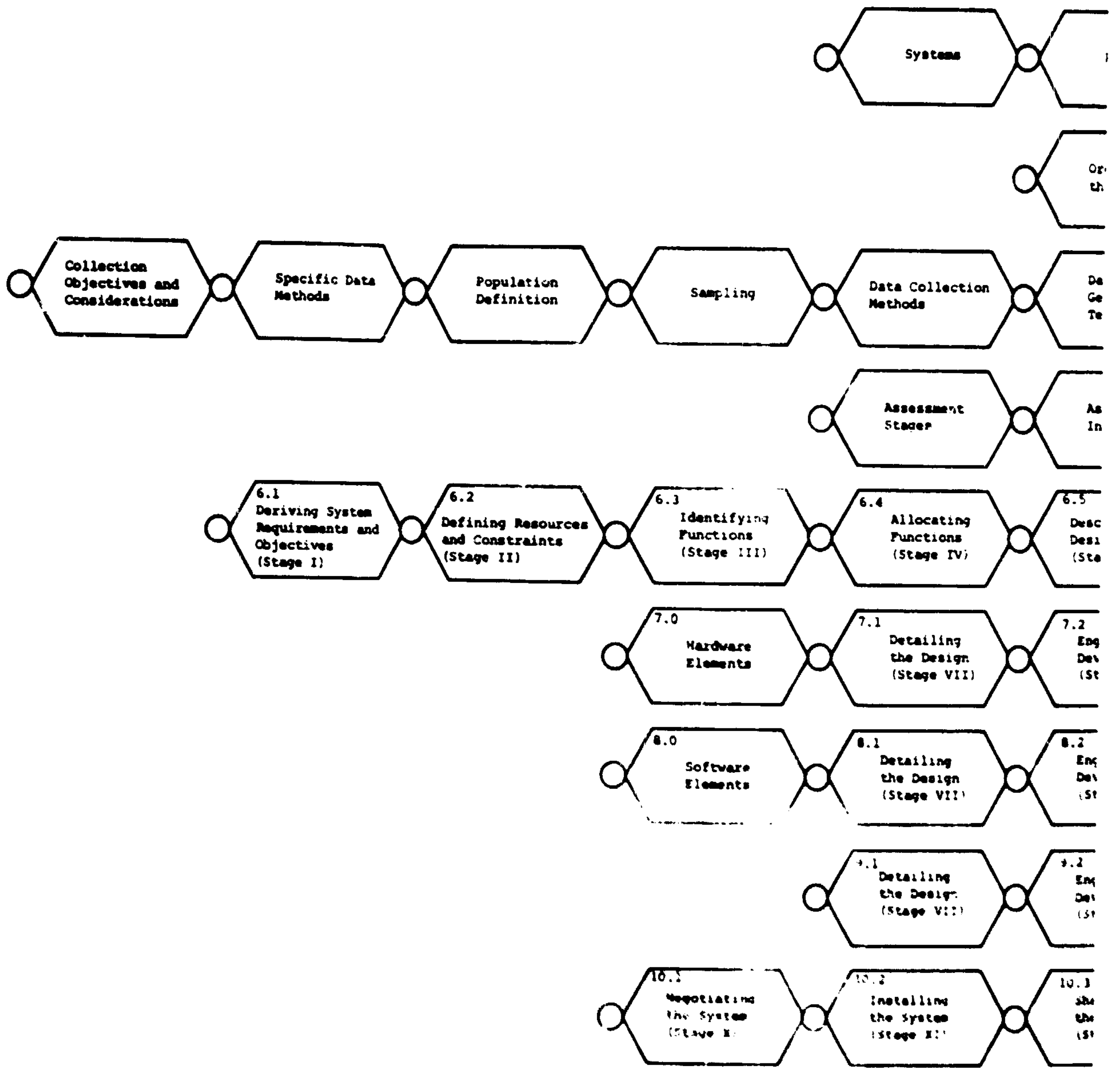
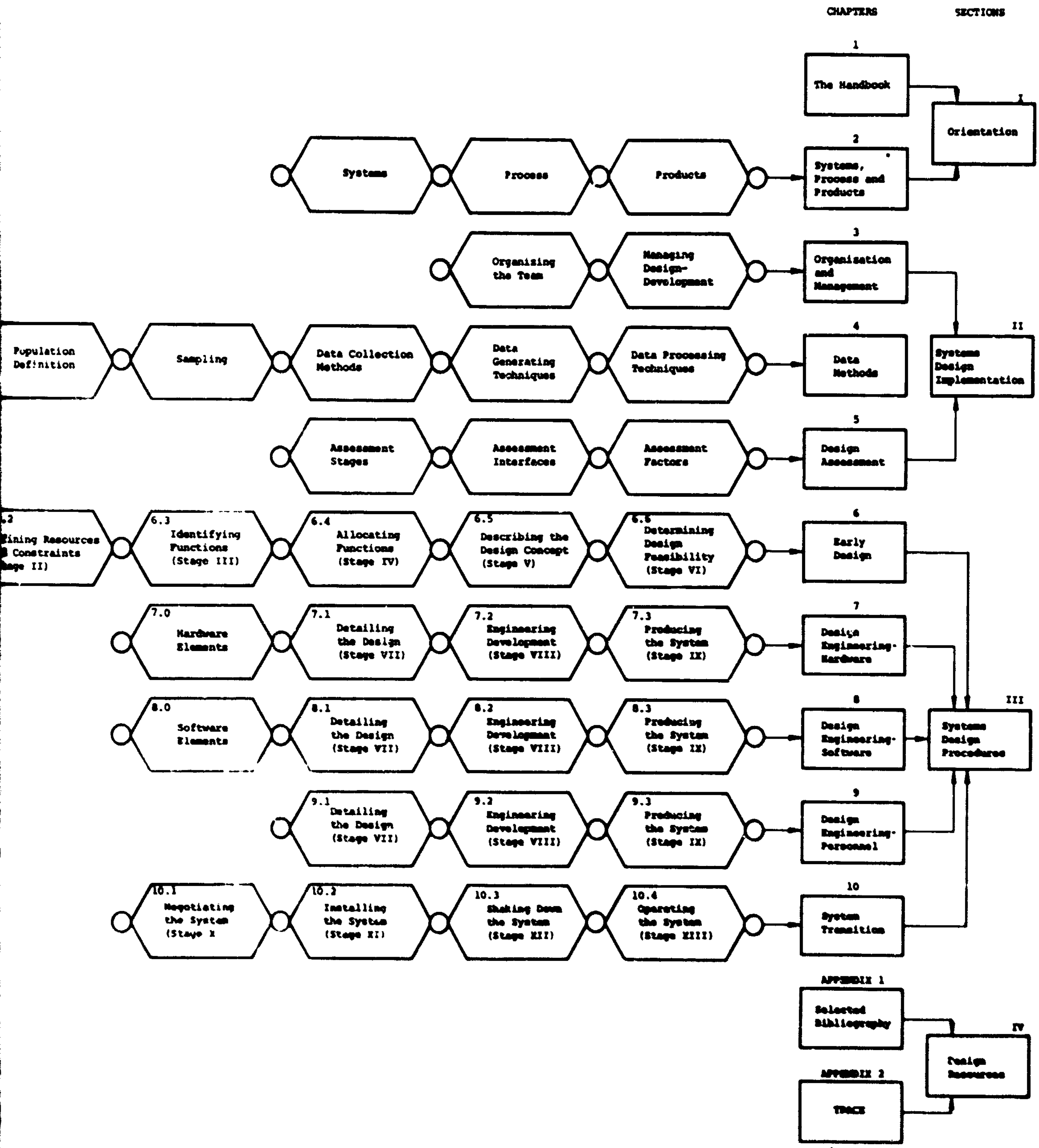


Figure 1-1. Overview of the Handbook



view of the Handbook

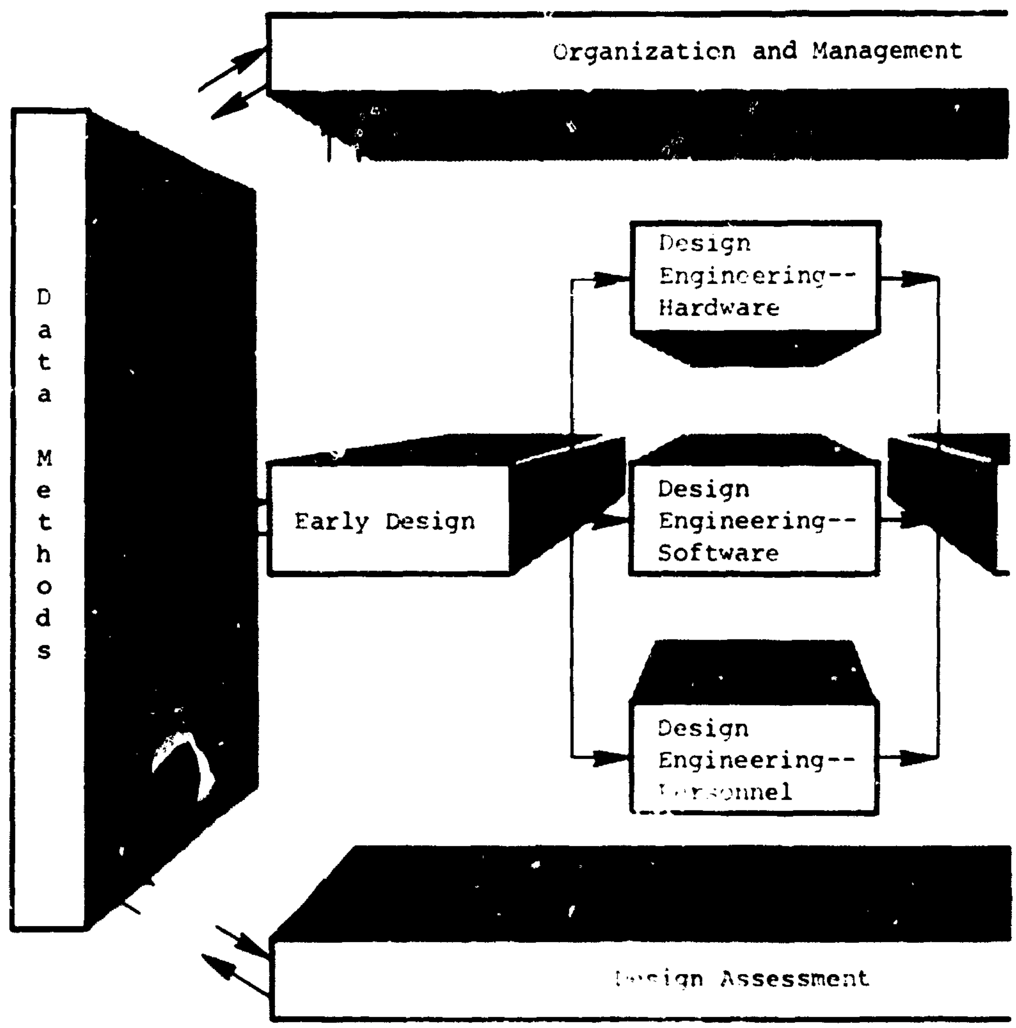
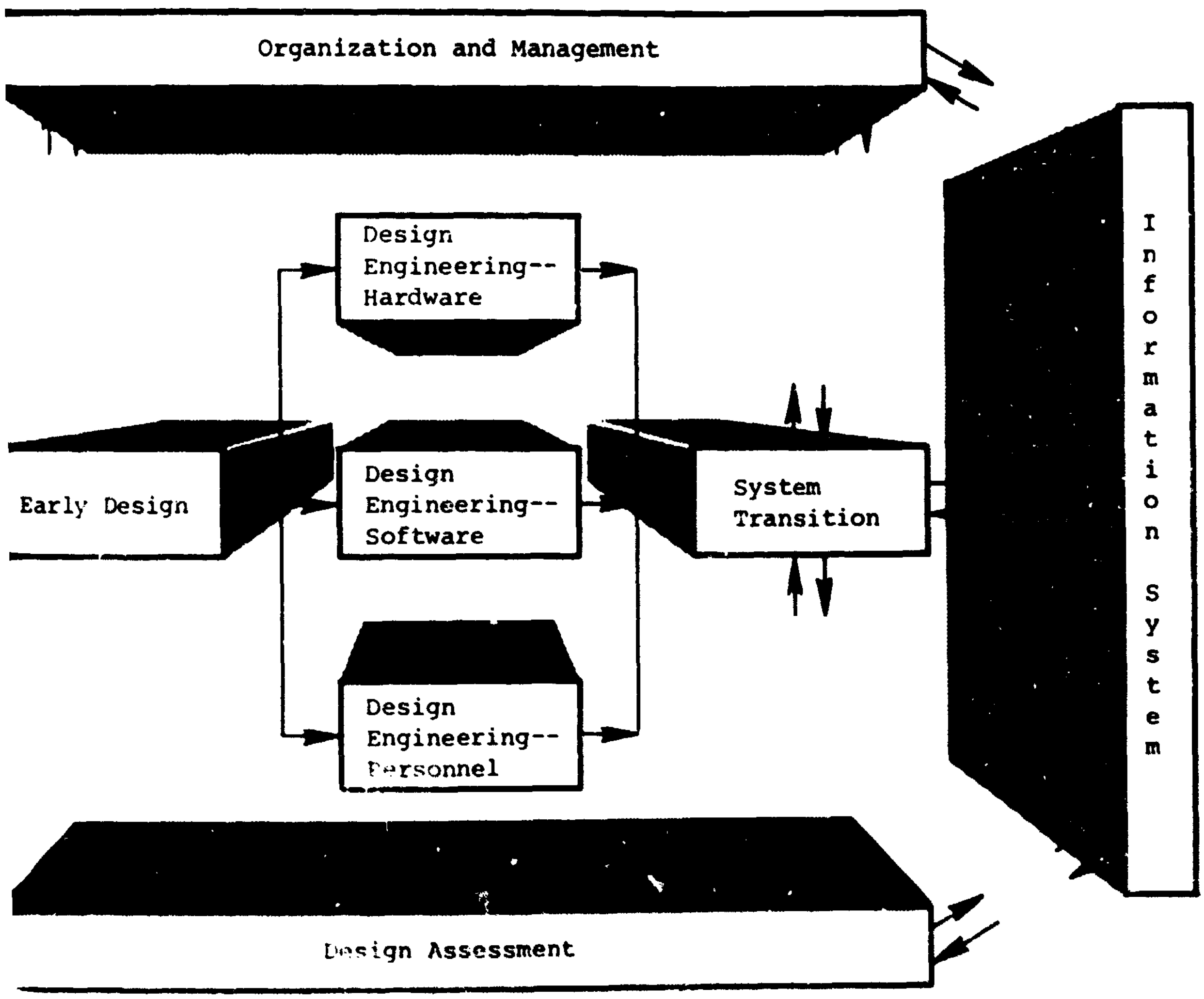


Figure 1-2. System Design and Development


Organization and Management
Design Engineering--Hardware
Early Design
Design Engineering--Software
System Transition
Design Engineering--Personnel
Information System
Design Assessment

12

CHAPTER 2

SYSTEMS, PROCESS, AND PRODUCTS

In all probability, undertaking the first or any new information system analysis and design problem will seem much like confronting a bowl of wet noodles. On the one hand, prospects for untangling matters are scarce and complicated. While on the other, loose sticky issues spring up for consideration on all sides. Clearly, some insight into the overall nature of this game is necessary in order to become a player. Hopefully, access to a "big picture" will also protect against becoming lost in the details as you carry out the system development project.

The overview is developed in three parts. Systems answers the question: What is an information system "thing?" It discusses the chief identifying characteristics and design features which mark this class of system; in general, these are shown to be responsive abstractions of crucial user requirements. Process outlines a coherent strategy for gathering pertinent design data, synthesizing a design, and implementing an operational capability in the user's environment. It covers where to begin, how to proceed, and what to consider. Products describes intermediate results of the development process as successive snapshots of the evolving end items. In so doing, it points out the formal and informal products used to create the system, keep development on course, and inform both management and user of progress.

Systems

Definition

Information systems are purposeful organizations of hardware, software, and personnel components which accept, manipulate, and disseminate information.* An effective information system informs its designated users on the

* Information is any intelligible representation of fact or circumstance (such as a message, symbol set, or graphic data) which affects the recipient's perception of the state of some phenomenon.

state of specified phenomena in relation to the users' informational needs or organizational roles. Hence, the criterion of systems effectiveness is the extent to which the system delivers relevant facts and data to its users. Relevancy (of delivered information) is best measured in objective, quantitative terms as a function of the accuracy, completeness, and/or timeliness with which user needs are met.*

Figure 2-1 expresses the definition as an elementary single-thread model of system functions and first-order interrelationships. The representation is deliberately simple, in order to highlight the basic characteristics. For example, not shown are the duplicate functions ordinarily required in the real world to handle traffic load; nor are the iterative handling actions and feedback loops which implement data verification and query refinement. The point is that once the confusing overlay of application-specific elements is stripped away, a rather simple model suffices for the entire gamut of system possibilities.

Nevertheless, the nature and scope of application requirements appear to be virtually unlimited, as users increasingly demand access to all manner of natural and behavioral data. Information system design can and does involve just about every conceivable scientific-technical data collection, reduction, and presentation problem--fortunately, not all at the same time. Some idea of the range information systems take is illustrated by the following brief list of military applications:

1. Reconnaissance systems--airborne/ground-based target data collection and exploitation.

2. Early warning systems--missile/manned-aircraft penetration detection, identification, and tracking.

---

* User attitudes toward output--e.g., satisfaction or dissatisfaction with information received--are neither very precise nor reliable indicators of system effectiveness. The difficulty is that attitude formation is influenced by a large number of factors, not just those directly related to how well output matches actual need. Consequently, user attitudes are usually gross clues at best to system strengths or weaknesses; and the effort to pin down the underlying reasons may not be worth the cost.

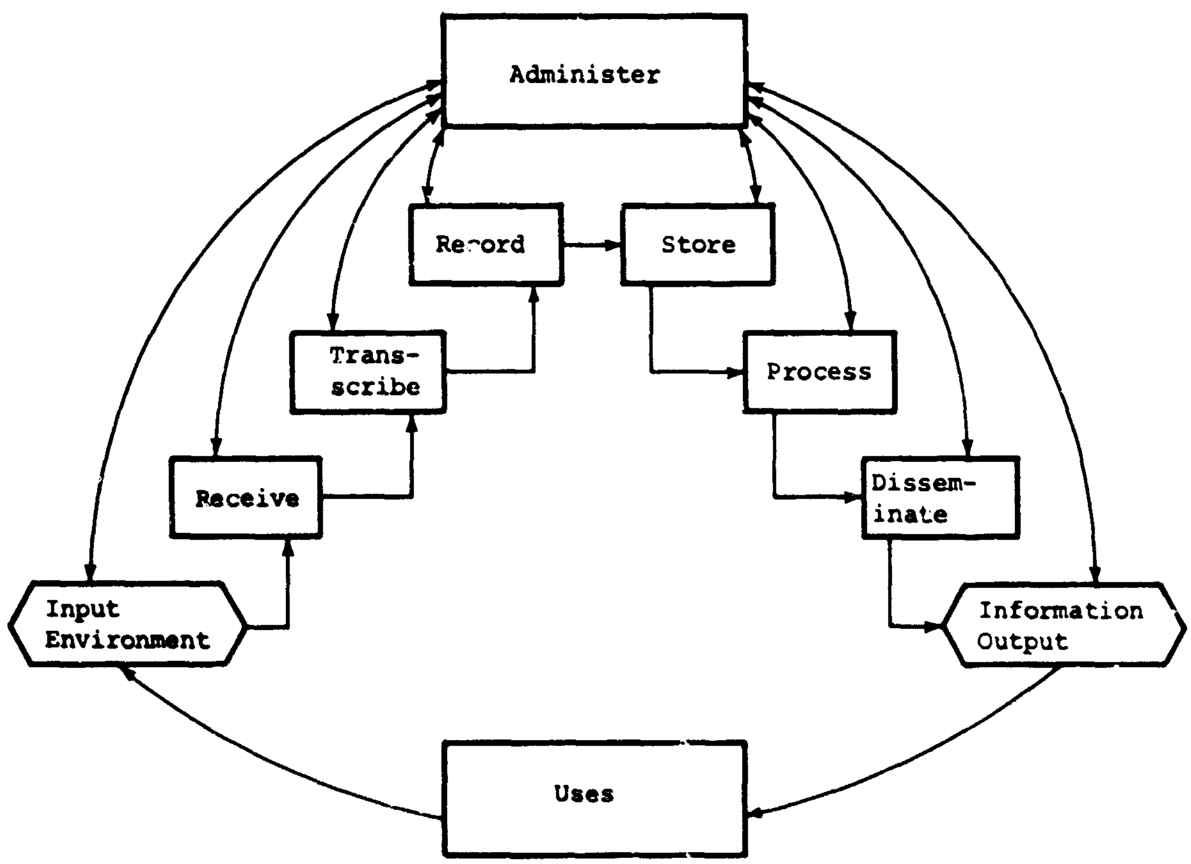


Figure 2-1. Information System Functions and Flow

3. Logistical support systems--material accounting and inventory control.

4. Air traffic control systems--aircraft take off, landing, routing, and navigation.

5. Weather observation-prediction systems--satellite/ground-based meteorological and climatological data collection and exploitation.

6. Scientific-technical information systems--documentary data collection, extraction, translation, and dissemination.

7. Personnel records systems--personnel legal, medical, historical data collection and application.

To conclude elaboration on identifying characteristics, Table 2-1 breaks out each model function to the next lower level. This level enables one to see the variety of application differences more clearly. (Subfunction labels may at first seem unusual, since these terms are meant to cover the entire range of information system applications, some of which may not be familiar to you.)

Design Features

The salient design and operating features characteristic of information systems provide further insight into design-development problems. Of the many recognizable features which hold for a small number of systems, seven appear reasonably universal.

1. Perhaps the single most important feature an information system must possess is responsiveness to the informational needs of its users. To achieve a responsive operational system is no easy matter, however. The key to success lies in anticipating that user needs change and providing for this in the system design. Furthermore, it is not sufficient to resolve this merely at the beginning of system operation. More important is how well the system

Table 2-1

Representative Information System Functions

PRIMARY FUNCTION	TYPICAL SUBFUNCTIONS	TYPICAL SUB-SUBFUNCTIONS
RECEIVE: Accept relevent inputs from the environment	• Monitor • Collect	• Scan • Track • Sense • Detect • Identify • Classify • Filter
TRANSCRIBE: Adapt, direct, and move data into the system	• Convert • Route • Transfer	• Sort • Modify • Encode • Decode • Assign
RECORD: Generate data analogs in recoverable form	• Read • Write	• Transduce • Index • Format
STORE: File data in a searchable medium	• Structure • Insert • Retrieve	• Read • Write • File • Search
PROCESS: Manipulate data	• Compute • Associate • Collate • Translate	• Add, subtract, etc. • Logical Sort • Compare • Sequence
DISSEMINATE: Distribute, present information to users	• Communicate • Distribute • Present	• Print • Display • Brief
ADMINISTER: Optimize and direct system operations	• Plan • Program • Regulate	• Set Goals • Allocate Resources • Measure Performance • Set Standards

functions accommodate these changing demands throughout its useful life. The utmost attention to accurate and complete requirements description is implied throughout development, but especially very early in design.

2. Information systems must cope with a dynamic source of input. Data of interest is embedded in a stream of non-pertinent bits and pieces of information. As a result, these systems must be equipped with capable receiving/detecting/filtering mechanisms.

3. The changing context from which inputs are withdrawn coupled with evolving user information requirements necessitates that systems also have the capacity to adapt. Designs tend to be open-ended and modular so as to permit increases (and decreases) in transaction capacity or shifts in content.

4. Information systems are commonly fitted with means for protecting and/or segregating data according to characteristics, nature of use, and/or user. The design and implementation of data protection/segregation features often raise unsurmountable difficulties for available technology.

5. The functions shown in Figure 2-1 form three interactive operating clusters, like links in a chain, which perform the familiar input-mediate-output actions. From a design viewpoint the requirements which define these three are quite different. Clearly, input functions must be designed around the nature and constraints of the data elements to be taken into the system, while output functions are influenced primarily by intended uses. Mediating functions are driven by whatever it takes to translate between input and output functions.

6. Data acted upon by information systems can only be described in terms of average parameters, not in precise quantitative terms. Design and operation must, therefore, pay special attention to data quality checks and controls throughout.

7. Current reliability/maintainability techniques are inadequate for design and operation of the software and personnel components of the system. Consequently, this area requires special attention also.

Process

More often than not, system development follows a path of trial and error, retraced steps, blind alleys and small advances. Even the most idealized representation of the process is considerably more complex than a single thread sequence of decisions. Perhaps, the most apt description is "progressive, goal-directed iteration," as pictured in Figure 2-2. It is progressive and goal-directed in the sense of definite movement toward end objectives; iterative in that tentative solutions are gradually refined through successive examination and revision. There is an interplay among initial problem definitions, context (resources, capabilities, constraints), and tentative solutions. Eventually, the candidate solution is assessed and, if found inadequate, the cycle is repeated. However, even an accepted solution is provisional, inasmuch as subsequent findings may bring to light possible improvements or an entirely new and superior answer.

Despite these realities, there is order in the development process. And, two convenient expressions of this order are used here to structure a discussion of the process, beginning with the earliest indication of a need for the system and terminating with full operation. The first divides the development effort into units we have chosen to call "stages"--i.e., pieces of effort which culminate in identifiable intermediate development goals. These discernible stages of development are the backbone of Chapters 6-10 (which cover the process in detail) and an overview of the process presented shortly.

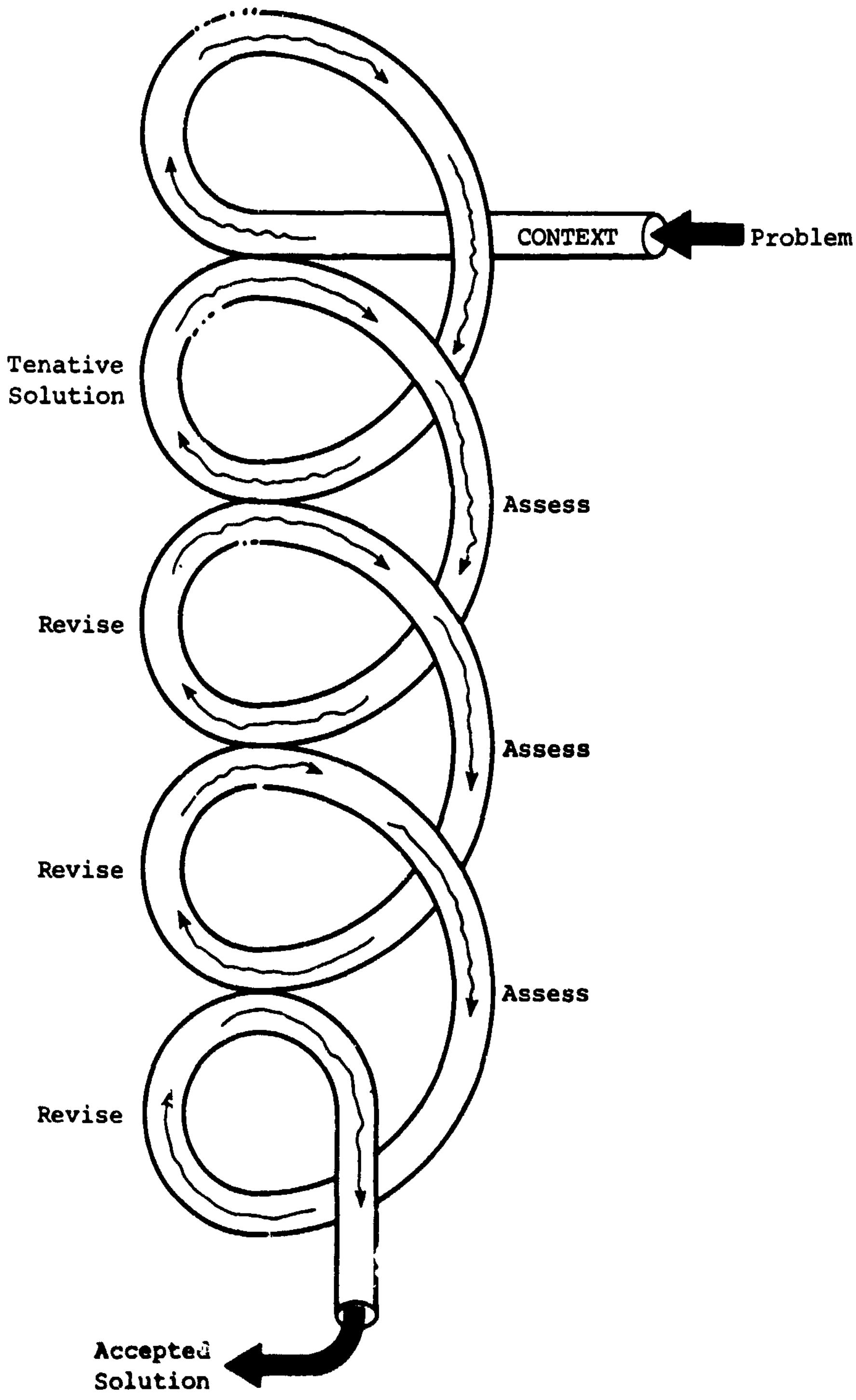


Figure 2-2. The Iterative Path of Design-Development

The second ordering is in terms of project milestones which denote significant administrative, managerial, and developmental events where timing of the action is the critical factor. As such, milestones are interwoven throughout the stages of development and may or may not coincide with stage terminal points. Major system development milestones are reviewed briefly later.

Stages

Figure 2-3 identifies the design-development stages used in this handbook to describe the general nature of system creation effort. The diagram also shows the organization of stages in Chapters 6-10 as well as the relationship of broader development considerations in Chapters 3, 4, and 5 to the central effort. Of course, development stages are nothing more than convenient abstractions which help to define, schedule, and measure progress in development. They derive their maximum significance when used in concert with milestones and other functional segmentations of the developmental process.

Stage I. Deriving System Requirements and Objectives. An information system can only be justified in terms of responsiveness to user needs. Consequently, unless these needs are translated into explicit requirements and the latter into unequivocal development objectives, a properly channeled project with a successful end product is most unlikely.

A great deal of worthwhile design work, having relevance to later stages, is accomplished as a by-product of this first step as well. The reason seems to be that skilled, creative design is largely a repertoire of solutions in search of the right problem. Thus, a careful statement of requirements and objectives (the problem) facilitates early solution identification. There are real dangers in "locking on" a solution at this time; however, if it is poorly thought through or good ideas are rejected, either much repair work will be needed or a suboptimized system will result downstream, and both outcomes are costly and wasteful.

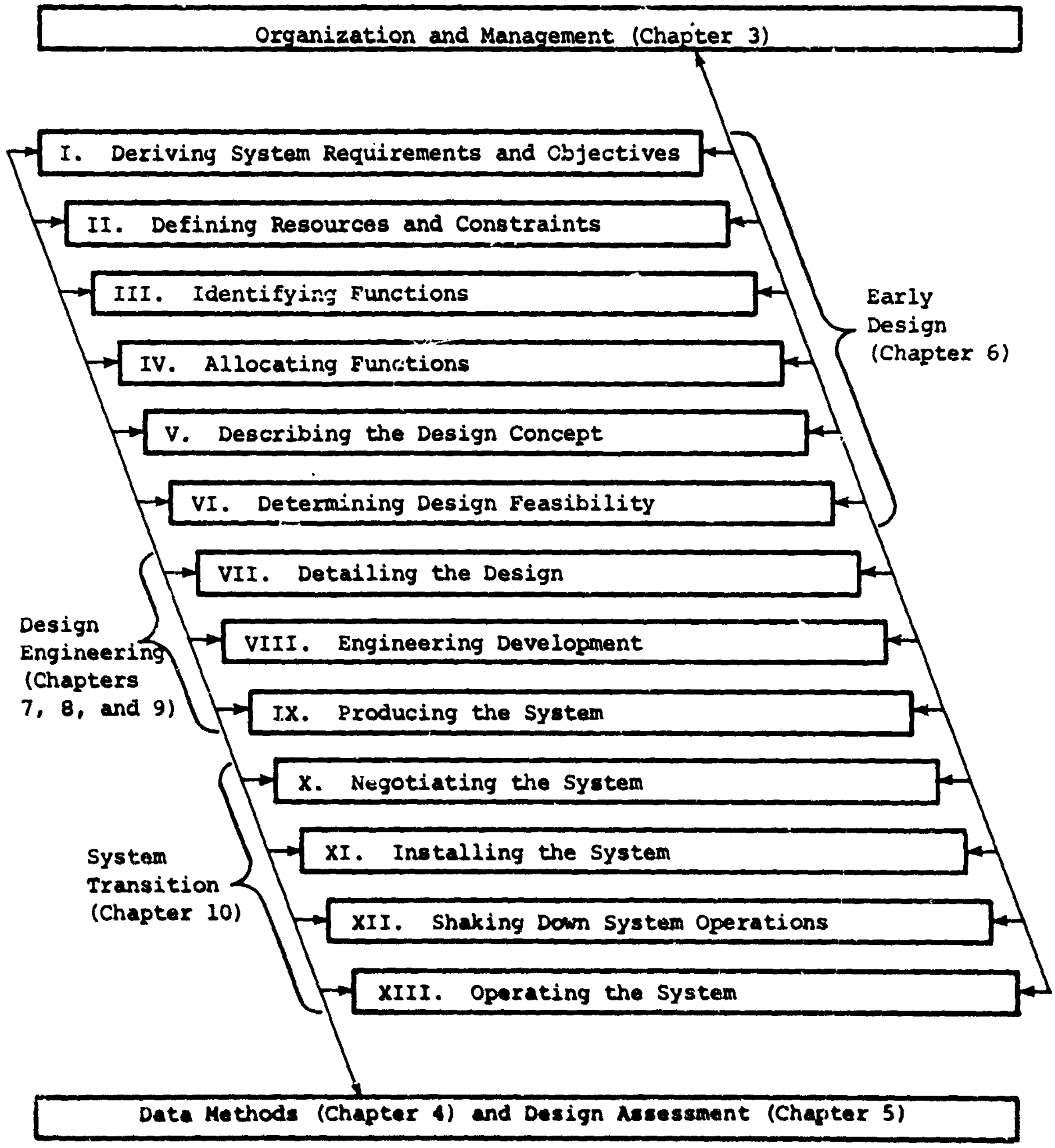


Figure 2-3. Stages of System Design and Development

Stage II. Defining Resources and Constraints. This activity concentrates on an accurate analysis and description of the contexts in which system development and operation must take place. Analyses emphasize factors and/or considerations which are assets, restrictions, or necessary accommodations.

Stage III. Identifying Functions. This activity aims at an initial, comprehensive, not too detailed description of system functions. Functions are those operations and performance capabilities required to generate system outputs (described under Stage I) from system inputs (described under Stage II). Functions description becomes successively more detailed as development progresses.

Stage IV. Allocating Functions. The functions defined in Stage III are tentatively allocated or assigned to hardware, software, personnel, or some mix of these components at this point. As functions are refined, more specific allocation or re-allocations are made to flesh out the system design.

Stage V. Describing the Design Concept. This stage represents the first comprehensive, formal attempt to describe the system design. The concept must consider relevant available technology, any advances required in technology, and estimates of operational results in connection with the system problem posed by user requirements and performance objectives.

Stage VI. Determining Design Feasibility. Studies of design feasibility are carried out at this time to determine: (1) That requirements/objectives and resources/constraints statements adequately describe the desired system; (2) that proposed functions and allocation choices are consistent with requirements, etc., and that no serious deficiencies exist; (3) that an optimal design has been achieved insofar as this can be evaluated. Feasibility determinations rely heavily, though not exclusively, on paper-and-pencil analysis, modeling techniques, and extrapolation from other system operating results. Increasing weight and attention is now given to qualitative, less easily measured factors which affect design suitability than formerly was the case.

Stage VII. Detailing the Design. This stage involves the reduction of identified functional capabilities to a specifiable mechanism such as a particular equipment, software routine, or personnel skill. The level of detail required is a direct function of the size and specifity of components sufficient to satisfy system requirements.

Stage VIII. Engineering Development. The objective of this stage is to generate the symbolic and/or physical representation of design specifications sufficient for demonstrating operational suitability. Since it may be impractical or unnecessary to demonstrate the entire system, subsystem/element performance may be integrated on rational bases to derive estimates of complete system capability.

Stage IX. Producing the System. This includes the procurement, fabrication, and assembly of the system. The extent of effort required varies greatly from one information system to another and from one component to another within the same system. For example, the major hardware elements are probably commercially available items and may even be already installed in the user's facilities; therefore, production would concentrate on unique applications software and providing personnel with new skills required.

Stage X. Negotiating the System. The period just prior to and overlapping system installation and shakedown usually involves intensive negotiations between the developers and users. The developer wants to pin down facility plans, installation details, test and evaluation procedures, etc., and hopes to pave the way for quick user acceptance. The user wants assurance that the system will be placed in an operational status as quickly as possible and, if shortcomings appear, that the developers are prepared to rectify deficiencies promptly so operations can begin. In short, the user's interests lie in delaying formal acceptance as long as possible, while obtaining operational use of the system; the developer's interests lie in a swift project wrap-up with a cordial user sign-off signifying operational take over and another successful development. Compromise is essential to resolve these rather antithetical ambitions.

Stage XI. Installing the System. Delivery, setup, checkout, and initial debug of all system components are the objectives at this stage.

Stage XII. Shaking Down System Operations. Takeover of operating responsibility for the system by the user is ordinarily initiated during this stage. The primary aims are to complete functional and operational acceptance tests, correct or, at least, identify any remaining design deficiences, and complete phaseover actions to the new system. Final acceptance of the system by the user is often contingent upon removing or alleviating the deficiencies uncovered during test and evaluation; as a result, the time period required to complete this stage may be as much as a year.

Stage XIII. Operating the System. This stage begins with formal user acceptance and ends with phaseout or replacement of the system. The obvious goal is full realization and exploitation of the operational capabilities provided by the system over the longest possible period of time. Updates and substantive improvements, particularly in software and access devices (such as interactive consoles) are common during the operational life of an information system; naturally, such changes tend to obscure estimates of system life expenctancy, but generally, the latter is about five to ten years.

Milestones

Figure 2-4 summarizes the major milestones associated with a system design and development effort. Virtually every milestone implies some administrative action on the part of development team members, user representatives, top management, or all three participants. These implications, shown on the right-hand side of the figure, are the important connectors between development stages and milestones.

Products

The products of system development are usually thought of as just the actual hardware, software, and documentation "deliverables" placed in the user's hands upon conclusion of the development process. For handbook purposes, a more inclusive definition is used--one that takes account of any result or action of the design and development process which is instrumental

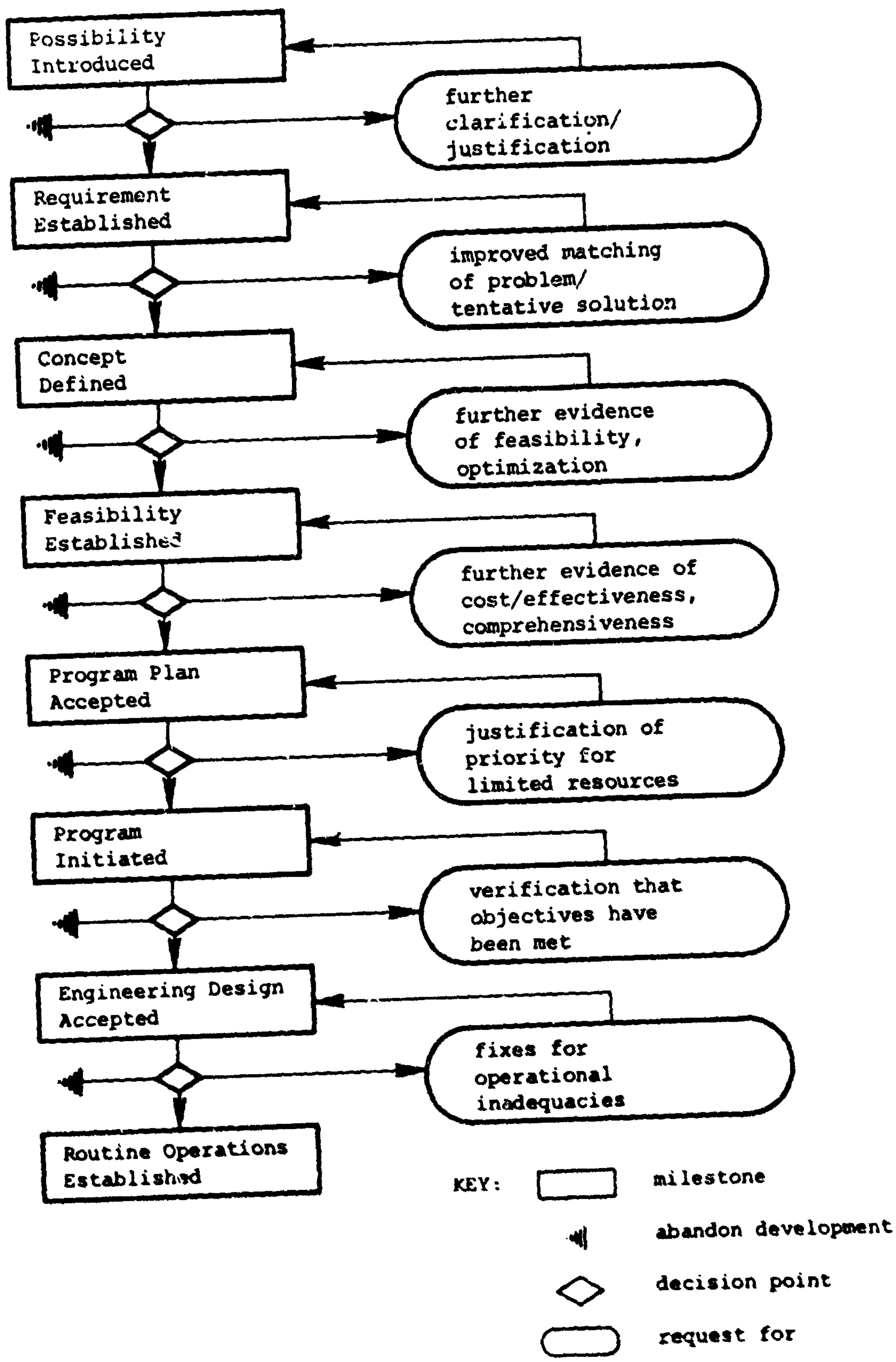


Figure 2-4. Major Design-Development Milestones

in reaching a satisfactory operational solution. We refer to those products and services--formal or informal--delivered by the design and development team to intended system users and for user-support groups. This focuses attention on the bases upon which users and their supporting elements (e.g., top-level operations management, logistic, and training functions) will judge the adequacy of the system. Figure 2-5 presents the identifiable formal and informal inputs; it also indicates their relationship to major milestones just discussed above. Of course, no one can describe in advance those intangible contributions to program success such as persuasive briefings and the like, even though these invariably occur and are often crucial.

Early Design Products

The products associated with early design stages and objectives are discussed according to: Those which contribute to establishing system requirements, and those which contribute to establishing the design concept and design feasibility.

Establishing the Requirement. Four products appear necessary in order for intended users and management, which legitimate requirements, to determine that an adequate set of requirements and objectives have been identified.

1. Developmental Policy Statements. The product documents in an objective, unambiguous, and concise a manner as possible the development rationale, the products anticipated, and preliminary evaluation criteria. Such statements would provide:

   a. Source(s) of request for developmental effort.

   b. Nature of the problem--reason or need for development.

   c. Major requirements and objectives to be met.

   d. Description (tentative) of how the system would meet those requirements and objectives identified thus far.

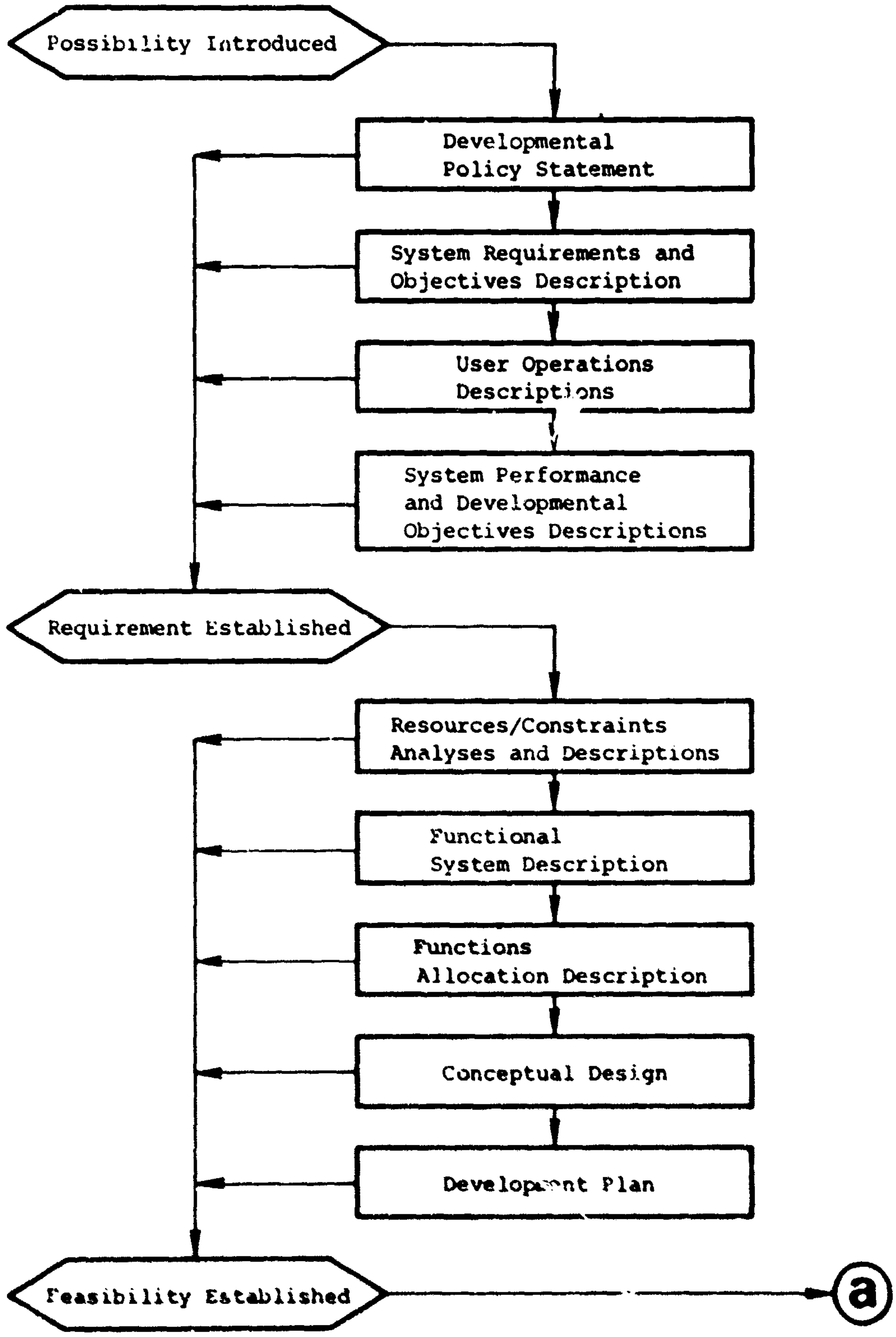


Figure 2-5. The Relationship of End Items to Major Milestones

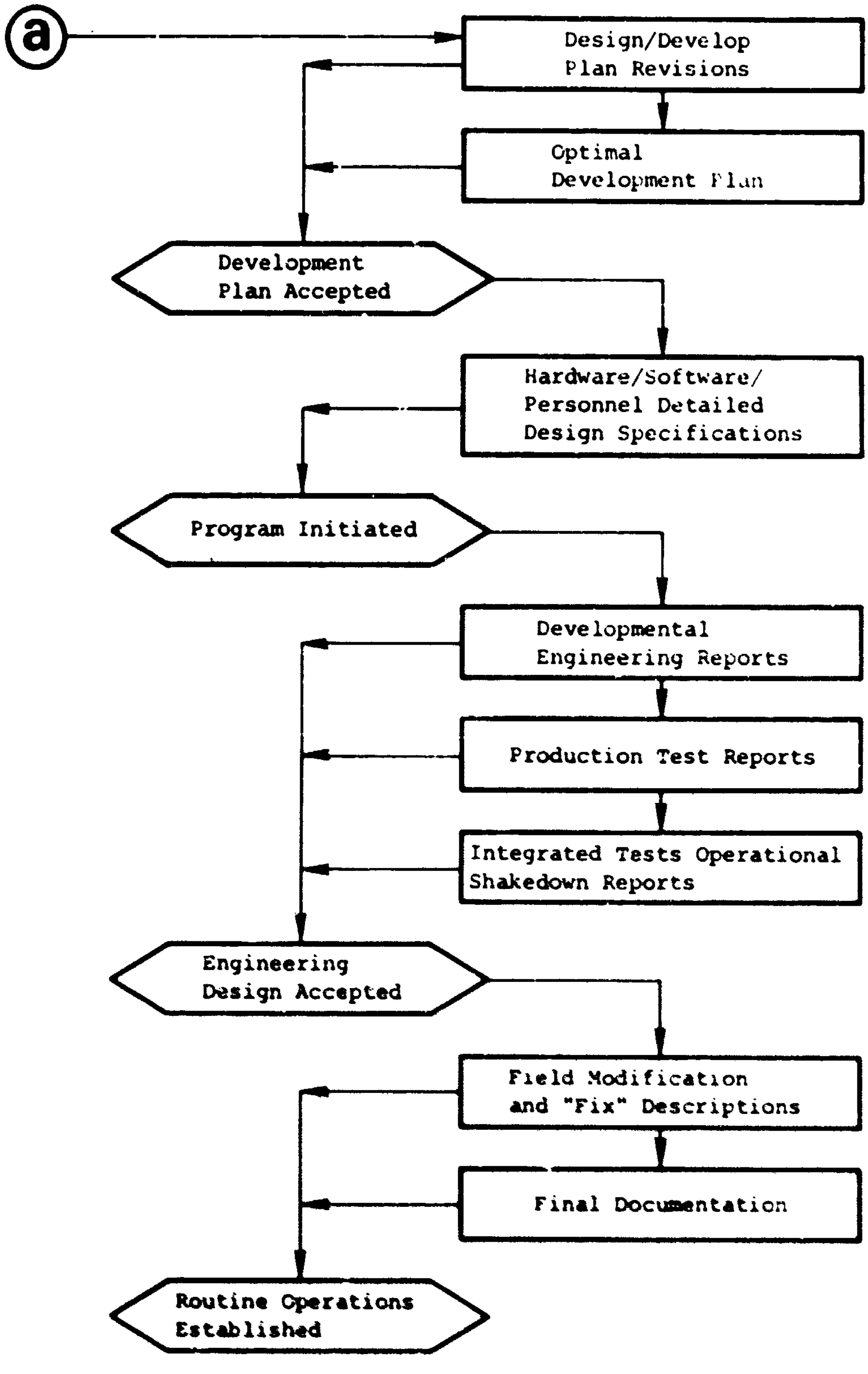


Figure 2-5. (Continued)

e. Consequences for intelligence operations of developing and/or not developing the capabilities.

f. Relationship to existing or planned capabilities.

g. Preliminary, broad, probable development time and costs.

h. Selection and justifications for developmental areas.

i. Tentative design-developmental objectives.

j. Performance criteria of significance to users.

2. System Requirements and General Objectives Descriptions.

This product summarizes the findings based on available information with respect to:

a. Scope and nature of requirements for each user or using element.

b. Alternative requirement sets considered in arriving at a "best fit" set of design objectives.

c. Assumptions and hypotheses underlying the "best fit."

d. Concise, unambiguous description of the general design objectives.

3. User Operations Descriptions. This product aims at generating a sufficiently detailed description of operations anticipated under the new system such that confirmation can be obtained from intended users. To this end, the description should identify at mission, function, and possible task level:

a. Continued system operations.

b. New or drastically modified operations.

c. Discontinued operations.

4. System Performance and Developmental Objectives Descriptions. This product provides a priority ordered set of system performance objectives and the implications which these hold for development action. These descriptions should:

   a. Permit evaluation of the adequacy of available resources with respect to the detailed objectives.

   b. Provide concise statements for subsequent formulation of the system development plan.

Establishing Design Feasibility. Once the constellation of required system outputs in the form of performance requirements and related developmental objectives has been established through user review and approval, the set of feasible (available or developable) inputs can be established.

Five aggregations of design and end products are minimally essential for an orderly progression to feasibility concurrence. While requirements establishment was aimed at successfully communicating system output characteristics to identified user groups, relevant feasibility determinations are communicated to those top management levels in control of development resources. For this purpose, no level of detail, depth of analysis, or skill in presentation may be too great.

1. Resources/Constraints Analysis and Description. This product comprises the results of quantitative and qualitative statements of resources according to the categories of development and operational system status. In general, the descriptions should offer a homogeneous set of parameters chosen on the basis of the informational content of the resource and constraint statements and the utility these parameters would have in demonstrating the extent of compatibility between resources/constraints and requirements/objectives. The extent of compatibility achieved is represented in the detailed tradeoff analyses reported on both

system operational and system developmental considerations. Quantification of the parameters and tradeoffs should be sought to the maximum extent possible.

2. Function Description. This product describes all input-process-output statements required to adequately represent the design. Since its primary user is the design team, the form and content may vary widely. Some utilization of system modeling techniques is common, however.

3. Function Allocation Description. This product presents the results of function allocation and should delineate results of allocation, evaluation criteria, and those instances in requirements and objectives where modified in order to achieve allocation. The resultant functional process statements organized according to function responsibility must be sufficient for: (a) assessing cost/effectiveness of the allocation decisions; (b) demonstrating comprehensive coverage of objectives and resources; (c) assessing effects on interfacing systems as well as those existing subsystems to be incorporated in system development.

4. Design Concept. This product presents the summary design description. It must facilitate responsive answers to questions aimed at how solidly the design will perform if developed to operational status.

5. Development Plan. This product describes a viable approach for development of the described capabilities. Such a plan should include:

   a. Job design--task requirements and position descriptions.

   b. Manning requirements and organizational structure.

c. Recruitment and selection requirements and techniques.

d. Training requirements and proficiency measurement techniques.

e. Group coordination and communication strategies.

f. Expected outcomes.

g. Scheduling.

h. Cost considerations.

i. Workspace layout and environment.

j. Job performance aids and equipment requirements.

Design, Engineering, and Production Products

The products generated during these stages include: functional specifications, engineering specifications, engineering drawings, and the hardware, software and personnel cadre elements. Chapters 7, 8, and 9 treat these products in detail. Furthermore, these products are covered thoroughly in terms of documentation requirements by existing United States Air Force program management regulations and directives.

Installation and System Shakedown Products

The products generated in connection with these two stages include: Facility layout diagrams, test and evaluation reports, reliability/maintainability reports, and special reports on system operating characteristics and/or deficiencies. Again, program management documentation spells out the formal requirements in detail.

SECTION II

SYSTEMS DESIGN IMPLEMENTATION

This section examines the essential activities which support system design and development and which occur continuously throughout all design and development stages. These support activities are categorized in three general areas which correspond to the three chapters comprising this section. A description of each chapter content follows:

Organization and Management (Chapter 3)--outlines the recurrent administrative activities which are necessary to organize and direct the system effort, describes the varied roles of management, and discusses the functions which management performs in system design and development. Consideration is also given to organization and management, including identification and selection, of the design team.

Data Methods (Chapter 4)--relates the methods and strategies generally relevant to the collection, manipulation, and interpretation of pertinent design data throughout the system effort.

Design Assessment (Chapter 5)--describes the approaches, considerations, and techniques employed by the design team as well as management to evaluate the emerging system as design and development progresses.

Figure II-1 depicts the relationship of these three support activities to design and development. The scope of this section is indicated by the emphasized portions of the diagram.

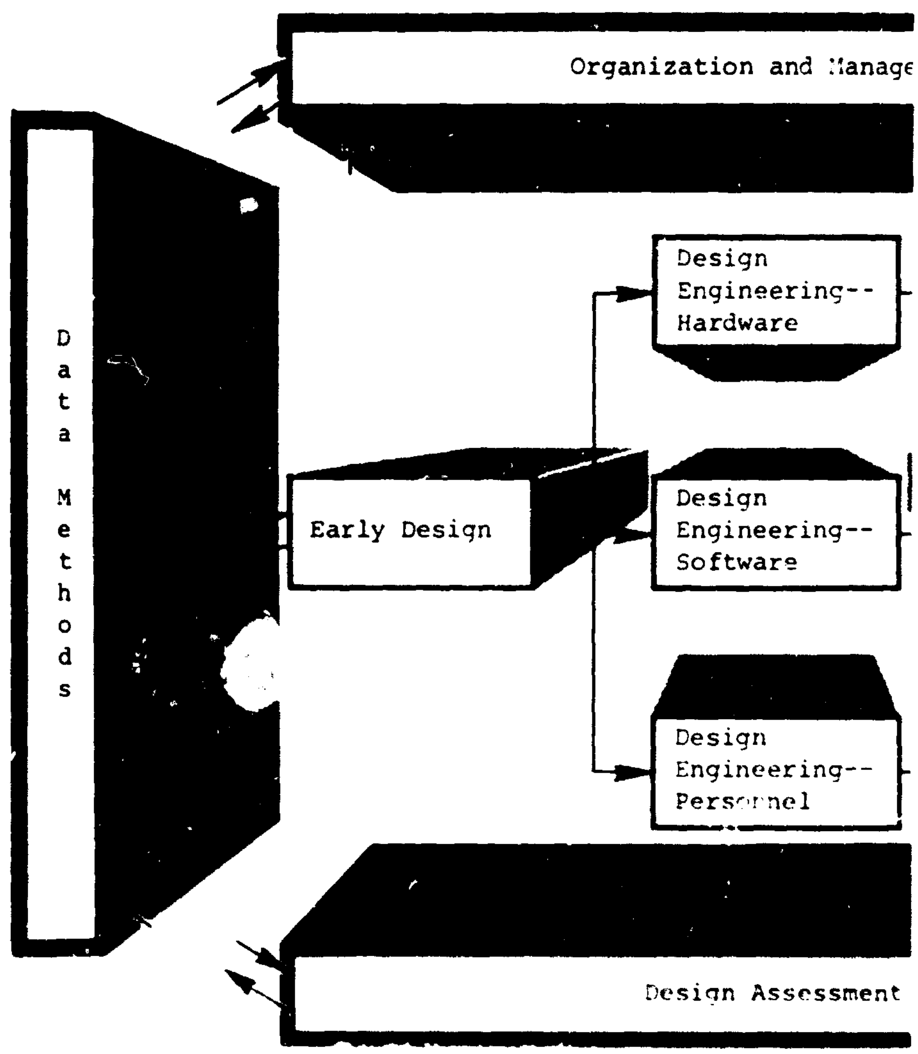
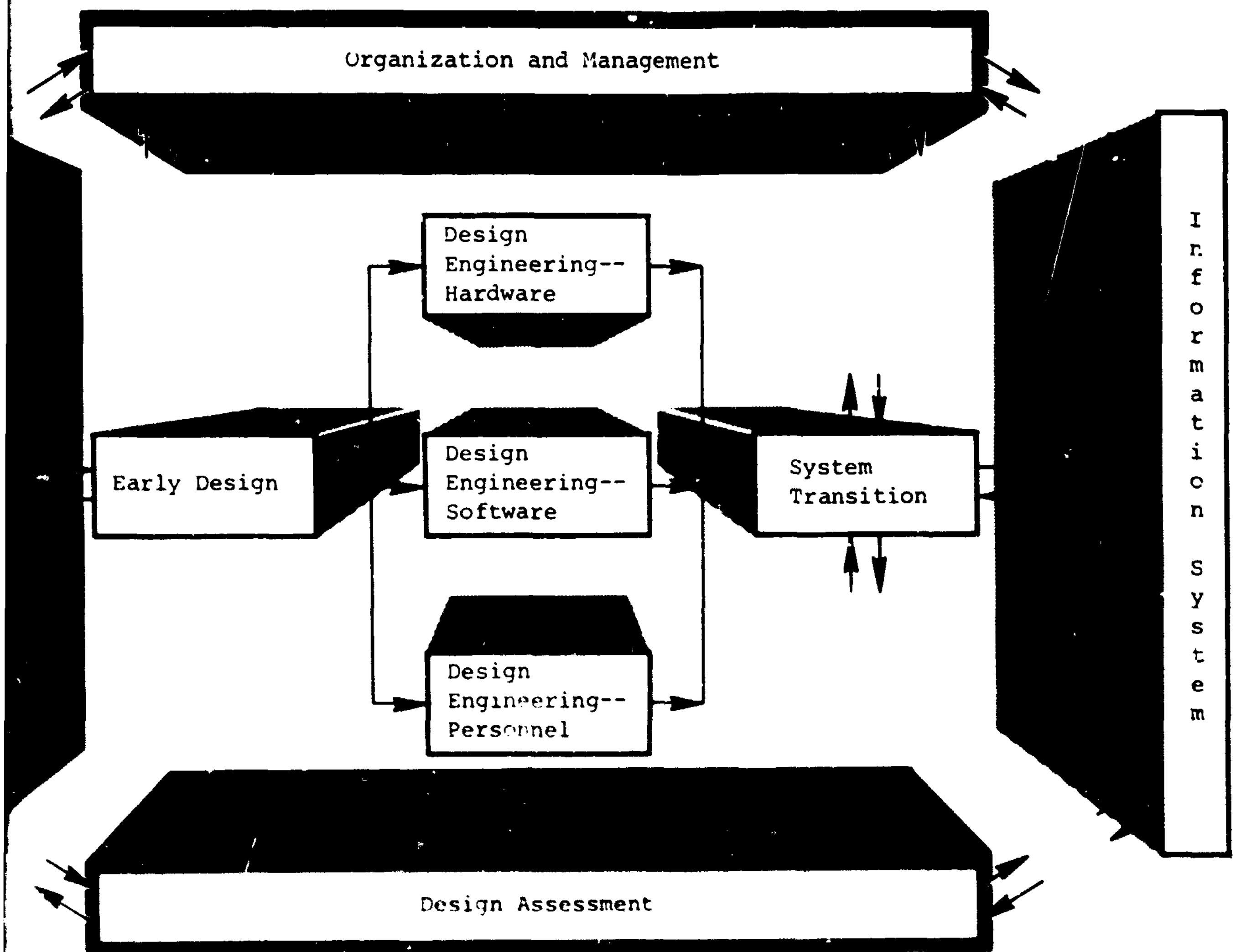


Figure II-1. System Design and Development



System Design and Development

CHAPTER 3

ORGANIZATION AND MANAGEMENT

Design-development management is not clearly distinct from other types: Its purpose is to increase and maintain coherence and direction in the organizational unit. But the manager's task is more complex in information system development, because these systems are usually more complicated. Thus, as manager of such a system building task, you must identify, pull together, and lead a diverse group of technological specialists. You must be knowledgeable in and comfortable with a broad spectrum of user applications, wherein--as we have already seen--the accurate determination of the user's requirements is especially critical. And, you must proceed against a backdrop of rapidly changing technology which has produced a bewildering array of components to choose from and very high monetary stakes, indeed. For these reasons, you should be aware of the important issues in organizing and managing this type of project. Hence, the discussion which follows is ordered according to considerations in organizing the team, then, to matters of managing successfully.

Organizing the Team

Team Composition

Probably, the first question immediately following establishment of a development project and naming of a project manager is: Who should be involved? A practical answer rests on the principal functions which team members must perform as shown graphically in Figure 3-1. Several points can be made in this connection.

1. To a large degree essential team functions cut across the scientific-technical disciplines involved. This fact strongly argues that team members be selected who can wear multiple hats competently. Moreover, the name of the game

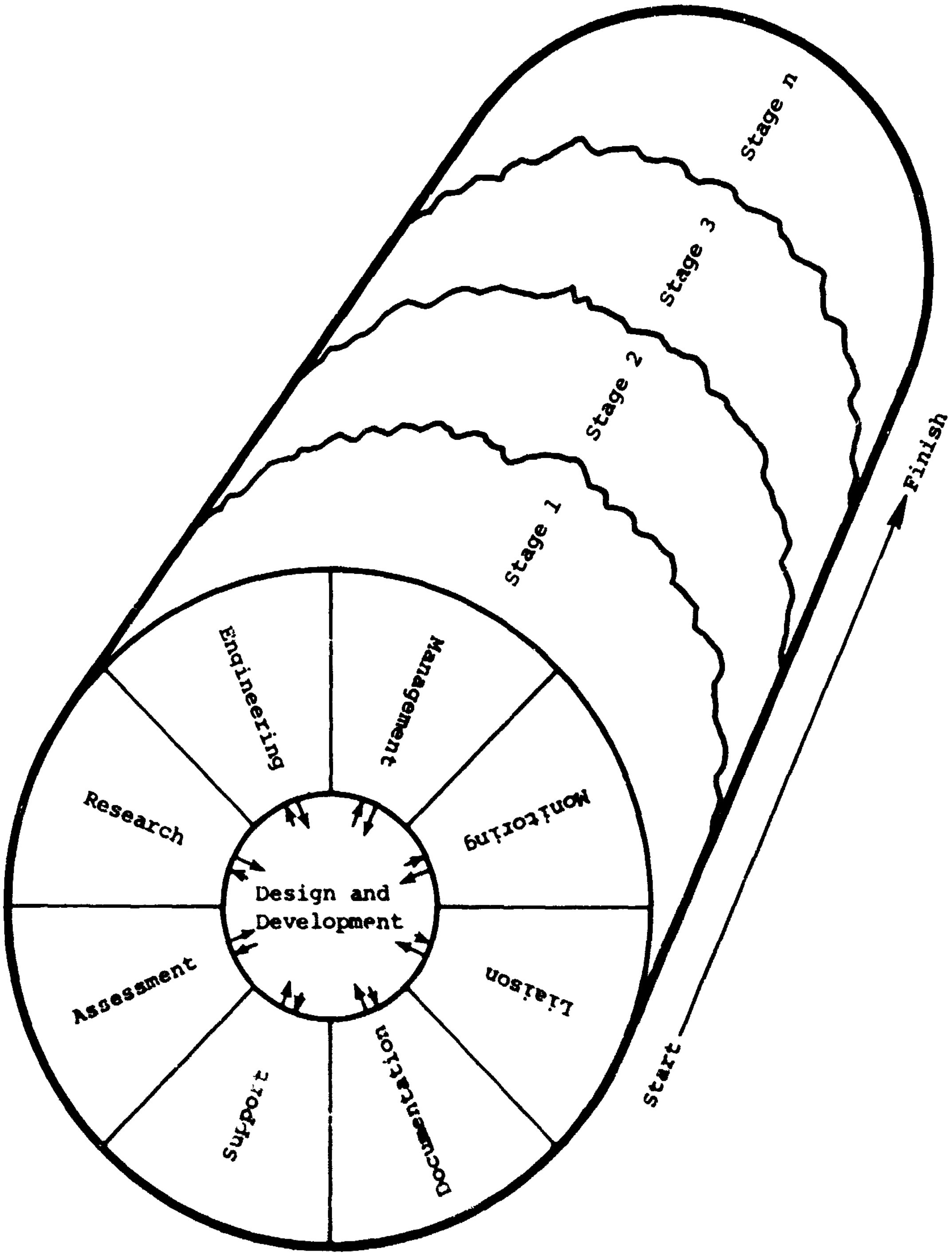


Figure 3-1. Principal Functions in Design and Development

in the early design stages should be to keep the team small and closely knit--a second consideration which underscores the first.

2. The diagram stresses that the skills, knowledge, and experience required for team functions be present from the outset. Consequently, it is important to select individuals who can, or will come "on board" at the beginning and stick with the project. Late arrivals or early bow-outs (necessitating replacement) cause too much backing-and-filling on the part of the manager as well as other team members. The same consideration holds for selecting user agency representatives, although you will obviously have far less control over the choice of an individual, time of arrival, and length of stay. (In fact, user representatives are most often located remote from the R&D agency which means, at best, brief periods of direct participation on their part, spaced throughout development). In addition, the amount of total effort in later stages of development is very much a matter of whether or not the exceptional efforts required of the user's personnel to accommodate the unfamiliar demands and shakedown requirements of the new system are counted. Thus, in the final stages, there may be an illusion of minimal effort when the user is actually expending monumental efforts to bring the new system up to operational status.

3. It is crucial that team members be selected for demonstrated skills in the intellectual processes symbolized in Figure 3-2, at least during the early design stages. While true that downstream efforts can compensate for defective work within limits, the costs are almost always greater than those incurred by a sound initial approach. Similarly, team members must be able to function effectively together in an informal, flexible, and relatively non-differentiated

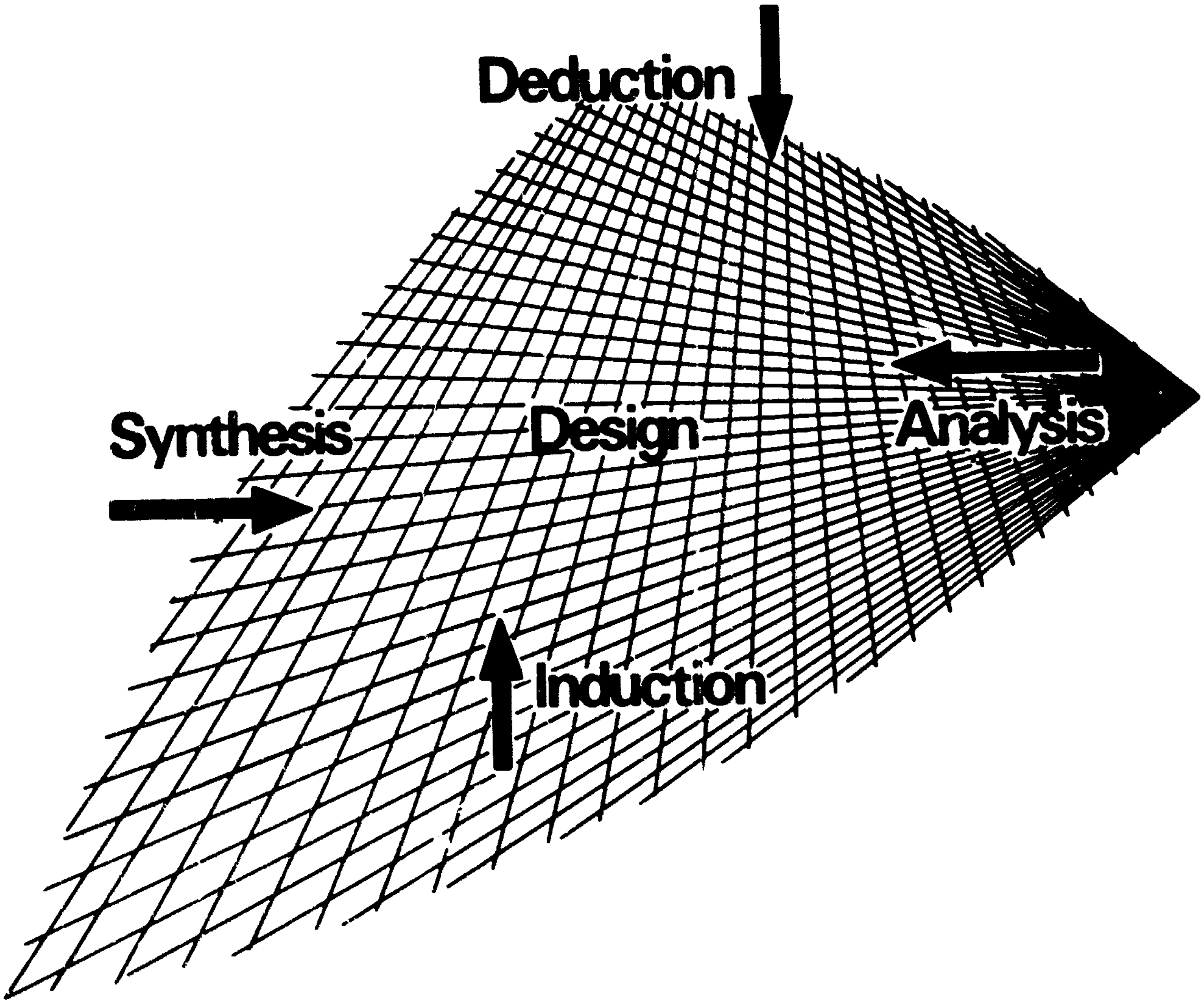


Figure 3-2. Design and Intellectual Processes

work situation at the outset, even though individual and group responsibilities grow increasingly formal and specialized as development progresses.

4. Although capability to perform Figure 3-1 functions may appear virtually equal in importance, in practice, certain abilities are more equal than others. Generally, for team members these are the abilities to: (a) manage time/effort effectively; (b) document or communicate; (c) liaison, particularly with user agencies and representatives; and (d) conduct evaluations or assessments.

Team Structure

Essentially, there are only two organizational structures available to the project manager of substantial system developments: matrix and vertical. Matrix is the most common, since it is prescribed Department of Defense (DOD) policy. In this case, the project structure is superimposed on the development agency's existing organization. That is, while the project manager is specifically assigned within the lead (responsible) agency element, most or all project team members are designated from among other elements presumably able to provide the various necessary technical skills. Administratively, of course, designated team members remain tied to their parent elements and project responsibilities represent additional duties shared with normal or other project duties.

In the vertical organization, a separate entity is created for project/program management purposes and ordinarily restricted to large-scale system developments. Here, the manager and most, if not all, staff members are assigned on a full-time basis; however, technical specialists may still be drawn from outside the development agency element to provide additional support.

Since the project manager usually has no option as to which structure used, his main interest is the major strengths and weaknesses of each approach. The strengths of the matrix structure are:

1. It is more quickly staffed.

2. Scarce technical skills are more readily available through sharing.

3. It is more easily fitted within existing development agency organizational structures, due to a common functional nature.

Matrix weaknesses are:

1. Too many layers of management above technically responsible project managers, with expanding reporting demands.

2. Project team members report elsewhere rather t[illegible] to project manager.

3. Authority is dispersed and may be very unclear.

Vertical strengths are:

1. It has the most successful record in complex projects.

2. Authority is more clearly focused in the project manager.

3. Project members are full-time and provide greater continuity throughout the project life-cycle.

Vertical weaknesses [illegible]:

1. For all but the largest most visible projects, it also suffers from too many management layers.

2. Competent technical development personnel are a scarce commodity, and there may not be a sufficient supply for all such projects at any given time period.

3. The number of staff members tends to balloon and go beyond the span of project manager control.

Managing Design-Development

For purposes of this discussion managerial processes are described in terms of roles, functions, derived activities, and resultant styles. The

relationships assumed here between the design-development process and the managerial process are presented in Figure 3-3.

Roles

Traditionally, managers are thought of in an operational role, acting as a source of direction and feedback to insure that workers correctly perceive and carry out assigned tasks. This managerial role predominates where most of the work force is involved in repetitive and relatively stable duties.

However, an accelerating technical and social rate of change has brought the maintenance and planning roles of management into prominence. In maintenance the manager is an analyst who insures the organization is in good working condition, detects when performance is not according to expectations, predicts incipient failures, diagnoses where corrective resources must be assigned, and judges effective problem solutions. In the planning role emphasis is upon being prepared for the future. A common planning error is to assume that most factors remain constant and that only a few, more unstable variables change. Sophisticated planning goes beyond the prediction of future status except in the broadest sense. Instead, the planner develops strategies and resources for flexible response to contingencies as these become predictable or occur.

Development project management derives from the general management roles of operation, maintenance, and planning--it also includes these roles. Notable relationships of development managerial roles to general management are suggested in Table 3-1.

Functions

Managerial functions are defined as sets of actions related by content, purpose, interdependence, or mechanisms of accomplishment. Three prime managerial functions are considered below:

Information Processor/Handler. The manager may be linked to an information processor/handler, and in this capacity, there are several relationships to which you should attend.

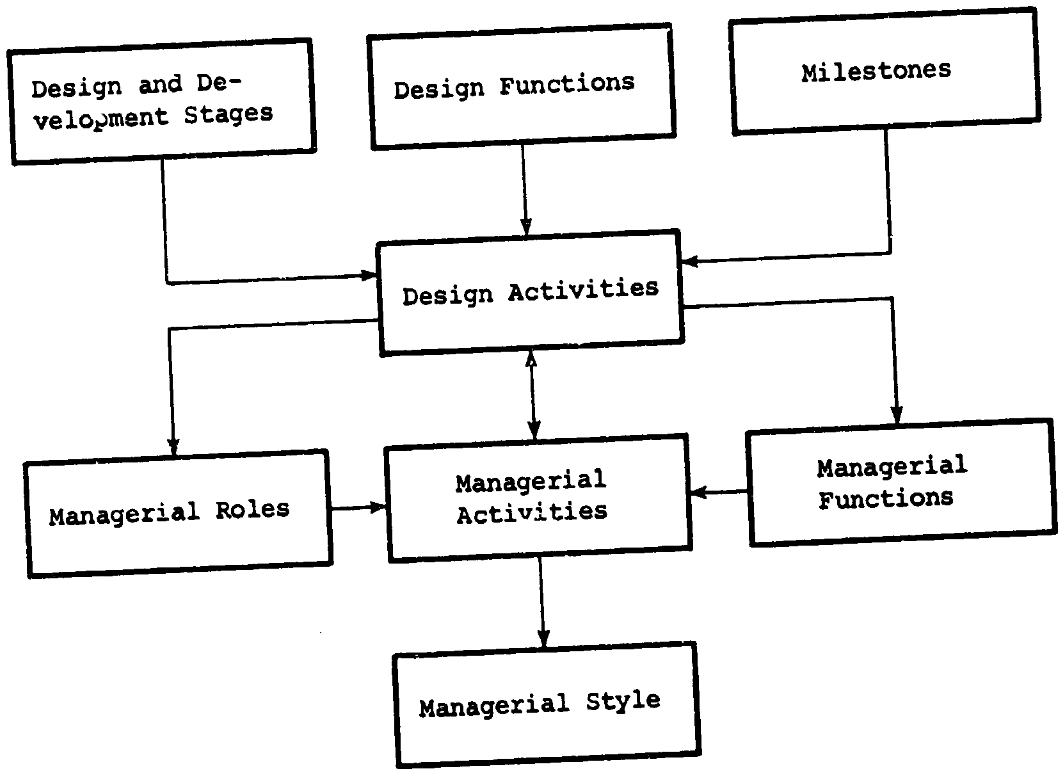


Figure 3-3. Design-Development and Managerial Process Relationships

Table 3-1. Relationships of Design-Development Managerial Roles to General Management

Design-Development Managerial Roles	Relationships to General Management
Operating	Design operations can interfere with activities under general management in a number of ways--the legitimate effort to obtain information about current operations and requirements for the new system may be unnecessarily disruptive, unreasonable demands for operating personnel and facilities in test may be made, conversion may be inadequately conceived, the initial operating capability for the new system may be unevenly provided. It is the responsibility of design management, and especially the project manager, to minimize such disruptive interactions.
Maintaining	Design for maintainability of the new system obviously has a direct effect on demands placed upon general management in order to maintain standards when the new system comes into the inventory. It is the responsibility of design management also to insure that design activities and conversion do not degrade general operations. In its effort to maintain adequate quality and pace for the design effort, design management may appropriately make unanticipated supplementary requests to general management as contingencies demand.
Planning	The planning of design management interlocks, in part, directly with the plans of general management. The project manager's job derives from the acceptance by general management of the need for a new system within his span of responsibility. The product of the project manager and those who support his efforts is expected to find its acceptance in the domain of general management--when general management anticipates and plans its advent.

1. The project manager is the appropriate channel for exchange of information between the project and other key individuals such as users and higher levels of research and development management. In fact, there must be a free two-way movement of relevant information, or neither the purposes of project nor higher level management can be served. For example, top management is responsible for deciding when and for what reasons to proceed; the project manager must be aware of and understand these facts and inform development team personnel. Or, user personnel must be kept informed of design changes, otherwise misunderstandings about system characteristics develop, and these confusions may jeopardize final system acceptance by the user agency.

2. The project manager has a high obligation to transmit accurate information to higher levels of research and development management and to user organizations. At the same time, the manager who concentrates only on technical accuracy will not do his project justice. He is appropriately chief advocate for the system while it is in development. He should take principal responsibility for assuring that the timing, frequency, organization, and tone of communications relating to the system maximize its consideration in an environment where there is always strong competition for attention and resources. To be fully effective, he must be prepared to argue the merits of the system in situations of indifference or hostility. He must identify the key individuals who influence crucial decisions, regardless of their location on a formal organization chart. Within the constraints of allowable communications, he must feed appropriate information to these key individuals and be sensitive to feedback from them.

Evaluator. The project manager is an evaluator in a most important sense: He must have, evaluate, and verify information concerning the status

of the effort on a continuing basis. To do so, he requires an effective review mechanism.* Yet, one of the most difficult problems in systems development is timing, frequency, and form of managerial review and approval. In particular, the technical development work, although secondary to informed management decisions, should not be impeded by the review process. Figure 3-4 presents the nature of information flow for design review which reflects the following points relating to the manager's evaluative function:

1. There is an interlocking relationship between review and other aspects of the development process.

2. The technical staff, intimately involved and informed about the details of design, represents an essential buffer between on-going development and review. The issue is not whether that staff can or should be by-passed, but rather, how it can effectively and efficiently play its role in design review.

3. Effective review requires a differentiated and organized flow of information among different levels and types of management.

4. Positive feedback is required in the developmental situation; review in one phase is the source of specific developmental goals for the next phase.

5. Accumulated experience with review provides a potential source of information having applicability beyond the confines of the immediate situation.

6. Technical assessment is the basic source of information used by management for project evaluation, but all design information is potentially useful.

* Chapter 5, Design Assessment, summarizes evaluative techniques which can be applied to management information as well as to system design products.

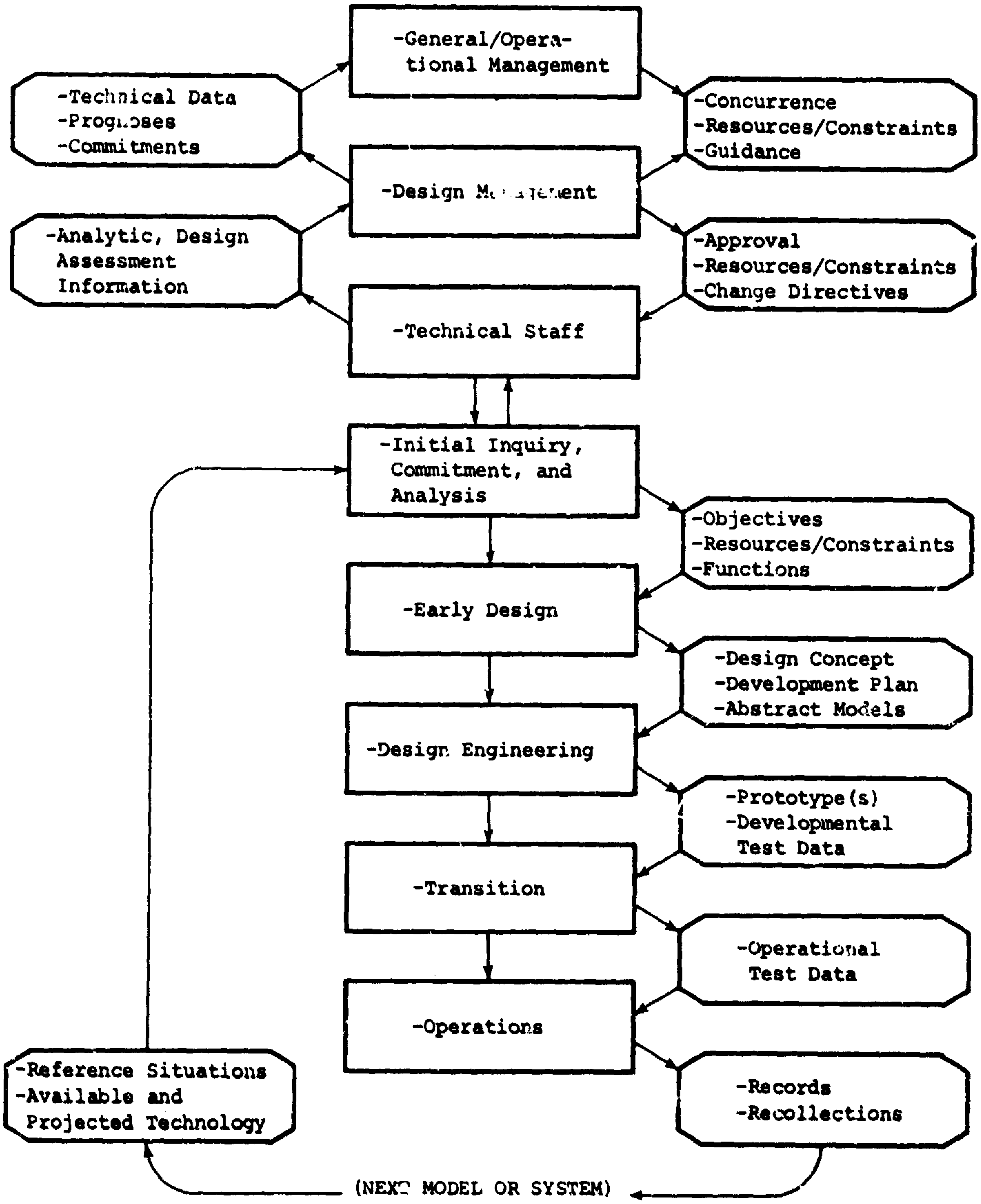


Figure 3-4. Information Flow for Design Review

At the same time, the manager must perform the evaluator function in the face of several ambiguities. Chief among these are:

1. Ambiguous management responsibilities. It is not clear that the project manager is responsible for total ongoing evaluation of system development, since he shares both responsibility and authority with procurement officers, higher level R&D, and user agency managers. In many instances, the lines of communication, which should focus on the project manager, do not; hence, he cannot properly pass information upward or receive sufficient information.

2. An ambiguous system development process. The development process is partially, at least, an improvisation--there is a general strategy, but no pre-programmable instructions for executing development. Therefore, at any given time, there is uncertainty due to shifting final objectives and requirements, constraints on available development resources, and limited techniques for quantifying such critical variables as system performance or requirements.

3. An ambiguous review process. Available review methods have not kept pace with the complexity of the development process which they are intended to reveal and control. Also, a general model is lacking to guide selection of evaluation criteria or basic data for different, specific review situations. How best to provide feedback from review actions to all concerned is also not well defined.

4. Ambiguous development phase relationships. The benefits from effective review decisions decrease with elapsed developmental time, while the quality of information available to support effective decisions increases. Also, the costs of premature decisions (in the form of sub-optimized end results and/or retro-design) and of delayed decisions (in wasted developmental costs and/or deferred benefits from an improved system) are difficult to estimate or balance.

Resource Allocator. The system development manager is an allocator of resources: Time, dollars, personnel, equipment, and facilities. Before treating each of these areas, there are some general allocation caveats which should be considered.

1. Allocation problems frequently arrive in the guise of a requirement to allocate a single resource. It is seldom that considerations can be so limited, however. Instead, you must deal with resources in interlocked bundles, rather than in optimized clusters of your choosing. For example, when development services are procured, the combination of problem approach, personnel, and facilities offered by a given contractor must usually be accepted, even though the personnel of one, the equipment of another, etc., is preferable. The important consideration is to factor in all the resources in making a choice, and not choose simply on the basis of the single most important or most obvious.

2. Available information on resources is likely to vary in accuracy and reliability from one to another and from time to time. For example, the time required for drafting a particular design drawing is quite predictable, but the time required to solve a given technical design problem is much less accurately predictable. Exercise care to insure that you do not either favor or penalize requirement areas on the basis of the accuracy or inherent uncertainty of available information. Rather, attempt to allocate resources fairly across requirements having varying degrees of definition.

3. You cannot expect to be handed resources for safekeeping until needed. More likely, you must squeeze every ounce out of the same sources who are demanding results. Furthermore, you are likely to encounter brigands bent upon appropriating your main resources, money, and outstanding personnel.

4. Paradoxically, system development involves extraordinary lead times to gain resources, but harsh demands to achieve instant results with them. Expect to be under many pressures to accept requirements and commit resources before a decent determination can be made. Learn early not to feel overly sorry when mousetrapped into "ballpark" estimates which suddenly become cast in concrete. Instead, make the best out of resources you can scrape together and be very realistic, even melodramatic about all those things which cannot be accomplished without adequate resources. Also, become sensitive to the use of "preliminary" resource requirement estimates.

5. Only the virginal design manager imagines having a completely free hand to define resources requirements or even allocate those nominally under his control. Rather, protect as many degrees of freedom as possible, exercise these wisely, and don't waste energy fighting windmills.

6. Formal cycles for approval of the resources needed to initiate a new system development or to provide continuity for existing efforts can be extremely slow. Happiness is finding resources which, although not specifically earmarked for the purpose, can legitimately be allocated for an urgent development requirement. Attend with uncommon sensitivity, memory, and skill to breaking loose contingency resources.

7. Few system projects have left resources of time or money, but many have left experienced design teams, equipment, and unique facilities. Look ahead and plan for the effective use of such legacies.

8. You must also allocate your own time and effort as project manager, and here, there is one completely worthless question you can ask, "Am I doing all that I should be doing

or would like to be doing?" Unless it is an utterly trivial project, the answer must be you are doing neither. If the project is of any size, the meaningful question is "What am I doing that someone else could do without causing dire consequences?" Stop doing all such tasks, forthwith. Finally,

a. If you spend most of the time putting out brush fires rather than selecting the tasks to emphasize, additional buffering is needed against the various firing lines.

b. If you are not dealing directly with user representatives on a regular basis, stop worrying about how the ship is running and start worrying about where it is going.

c. If there is any significant aspect of the system for which you are responsible but with which you are not reasonably current, stop competing with certain design specialists and work harder at being a manager.

At this point we return to the five resource allocation areas mentioned in the opening paragraph.

1. Allocating time and scheduling.

a. Recognize that it takes time to institute and achieve smooth operation of any scheduling procedure. Whatever the method used--PERT/Cost, etc.,--tailor the effort put into its use to the benefits realized in improved scheduling performance. In this regard the technique must permit flexibility at the outset of development and increasing precision as the project moves forward. Since there is actually

a hierarchy of milestones in every schedule, it is important to determine those which can be slipped without penalty and those which cannot.

b. Since design-development effort tends to expand and fill available time, avoid setting long-term deadlines without also setting intermediate goals. Generally, you should distinguish between informal goals for your own use--which can be relaxed--and formal assigned goals which are rather rigid.

c. No scheduling method is perfectly reliable; therefore, look beyond the built-in trouble indicators for other signs of potential difficulties.

d. Consider all scheduling costs, such as,

   1) Planning and estimating time/effort.

   2) Designing, monitoring, and updating schedules.

   3) Expending resources to meet scheduled events which would not have been necessary with a less stringent schedule.

   4) Eliminating design improvements that might have been desirable under less tightly scheduled activities.

2. <u>Allocating dollars</u>. Invariably, you will continually face a shortage of money resources to construct the operational system you would like to see developed and to support all developmental activities you would like to carry out.

Nevertheless, funds must be doled out to support the various required capabilities in such a way that minimum damage is done to the essential objectives. Perhaps the most difficult part of fund allocation is to put money where it is most urgently needed and still accomplish a balanced design which meets all critical operational requirements. It is a real temptation, particularly, to the technically oriented manager, to fund efforts where the state-of-the-art can be exploited and pushed farthest, even though the assigned mission is to develop a specific operational system and not to push state-of-the-art.

3. Allocating personnel. Except for one-man projects, you can expect most considerations discussed in Chapter 9 on personnel subsystem design to apply here. The similarity of personnel concerns for system development and for operating systems has increased as system development time has approached, and sometimes exceeded, useful system operating life. The outstanding differences between personnel problems for development and for operating systems are: The progressive change in roles for almost all personnel on a development project, the finite period of most development projects, and the somewhat unique tasks from one project to another. Another area of major difference is that the design team may have relatively little direct voice concerning the user organization, whereas the user agency may have considerable influence on organization of the project staff. A perennial personnel management problem is motivating team members to adhere to schedules. While avoidance of delinquency consequences is a strong deterrent for the individual or group that is constantly labeled as behind schedule, it soon loses motivational force. In this event rewards must be used to re-establish meeting schedules as important. Nevertheless, when serious hangups occur, you should, as project manager:

a. Identify the key figures responsible and bring them into effective communication for its resolution.

b. Press for full recognition of the sources of difficulty, refusing to accept trivial explanations just because they are comforting.

c. Keep everybody's sights glued to the realization that the purpose is to solve a significant development problem, nothing else.

d. Relentlessly hold feet to the fire until a workable resolution, involving all concerned parties, has been hammered out--ruthlessly denying quasi-solutions which make people feel better temporarily, but do not get at the heart of the matter.

4. Allocating equipment. The design manager must beg, buy, borrow, rent, or otherwise obtain computing, laboratory, testing, and other equipment required for efficient system development. His cost/effectiveness considerations must include choices among different items of equipment, different sources and modes for acquiring the equipment, and the costs of gaining access to equipment versus the costs of accomplishing design without such access.

5. Allocating facilities. Development projects are nearly always engaged in a game of "musical facilities" because of their relative impermanence, fluctuating size, and changing activities. The design manager must argue strenuously for minimally adequate facilities. Then the choice lies between living with a configuration of the project team that is out-of-date versus the costs of constant moving.

Managerial Activities and Styles

There is one final aspect of managerial performance which is important. This is style--that is:

1. The activities which you choose to emphasize--select those exerting a significant impact on development objectives.

2. The activities which you emphasize on the part of those who report to you--again, emphasize those things which relate importantly to development goals.

3. The freedom afforded subordinates--make end goals clear, but leave a great deal of freedom in how these are accomplished.

4. The concern shown for staff--while clearly concerned for success of the project, show interest in and support for the well-being and growth of staff members.

5. Use reinforcement--help the staff gain pride in their accomplishments. Do not spread phony enthusiasm, but take obvious note of, and pride in good work. Don't be afraid to be critical, but recognize that sharp criticism and punishment must be used with extreme discretion or it is simply disruptive, not appropriately motivational.

CHAPTER 4

DATA METHODS

Virtually all system developments place heavy demands on collecting, analyzing, and interpreting data for design purposes. This is certainly true of information systems--in fact, the tendency is to support every decision with data of some sort. Therefore, as a designer you must be conversant or, at least, very familiar with methods for resolving these demands which arise repeatedly throughout the course of development. You can also expect that conditions surrounding the overall development will dictate one of three data collection approaches. Namely:

1. Exploit readily available data, gathered <u>incidental</u> to some other effort--severe time and cost restaints override all other considerations, precluding the most austere tailored approach.

2. Identify and tap potential sources of <u>existing</u> data--time/cost considerations are less restrictive and a sound decision more critical.

3. Generate <u>new</u> data specific to the question at hand--time and cost are not controlling factors, insufficient data of any kind are available on which to base a decision, and the outcome is crucial to success of the system.

This discussion should familiarize you with the nature and scope of empirical methods for acquiring decision-relevant data under typical limiting conditions (primarily, alternatives 2 and 3 above). Where incidential data are the only resort, about all one can do is plan to be extremely cautious. Recognize also that the complexity of questions raised by design-development problems may require expert attention. Hopefully, this orientation will assist in identifying when and why such attention is needed.

Collection Objectives and Considerations

Assuming existing or new data are required to support design and development, you will have one of two objectives. One objective is to build a model of system operations. The other is to solve a specific problem or test a hypothesis about the system based on the system model. Hypothesis testing includes parameter estimations and prediction tests. Thus, a system model is a prerequisite to specific study or hypothesis testing. A model of system operations either exists or must be created from carefully collected, analyzed data.

A basis for distinguishing these two information gathering objectives is the difference in their corresponding data collection strategies. In model formulation, the data are used to create a total picture of system operations. The patterns, relationships, and structures of the data are emphasized. Data gathering strategies should be developed to reveal as many alternative views about system operations as possible so that the best single representation or picture can be constructed. This model must be referenced to the operational system as closely as possible. The model then serves as the framework for problem solving.

Information gathered for problem solving or hypothesis testing should also be referenced to the real operations of the system. The strategies used here should ensure that the characteristics of the data collected fit the model. That is, the model determines the degree of precision and reliability required for the data.

Aside from the particular objective of data collection and the corresponding strategies, you should be aware of two other considerations--the cost and effectiveness of your data methods. Cost is an important consideration in regard to the amount and quality of the data. Generally, the greater the quantity of data and the more accurate the collection techniques, the greater the cost. The degree of precision in data collection, measured by the above factors, should be evolved to avoid precision beyond needs which, of course, will translate into higher, probably unnecessary, costs.

Effectiveness of data methods refers to the technical characteristics of the data. Accuracy describes the general nature of these characteristics when analyzed in terms of several components: bias, precision, and level of confidence. Consider each component in judging the effectiveness of data collection plans.

1. Bias must be considered in sampling the population of data relevant to system operations. Explore the dimensions and broad ramifications of potentially biasing factors in collecting the data. Biases usually develop in connection with judgments about the range of variations in operations, key missions, and the effects of environmental factors.

2. Precision may be increased or decreased in two ways--by changing the number of measurements with a given data method or by altering the refinement of the method. Since there are no general rules for adjusting numbers or quality, you should evaluate tradeoffs between the two in terms of system specific factors and the cost considerations previously discussed. Two techniques which can be applied to guide precision aspects of data collection are sensitivity and sequential analyses.

3. The level of confidence for data should be established according to relevant factors rather than acceptance of an arbitrarily established value. The major factor you should consider in evaluating possible significance levels is the risk or cost of over- or underestimating and, thereby, falsely accepting or rejecting data. Decision theory provides a basis for judging conventional levels of confidence.

Specific Data Methods

Figure 4-1 is an overview of data methods. As shown, data sources are discussed primarily under the first method, population definition. The next

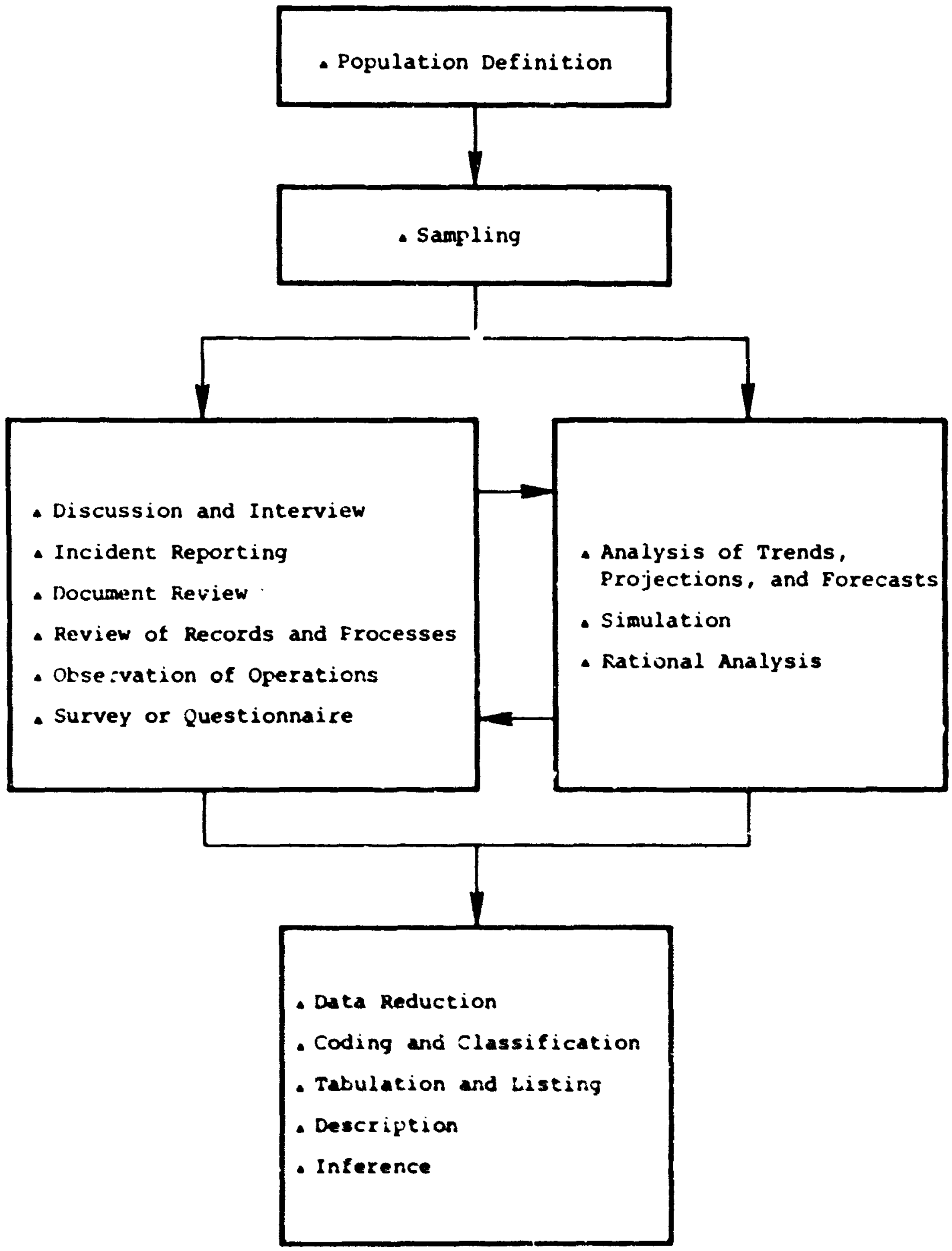


Figure 4-1. Overview of Data Methods

section, sampling, suggests some of the principal considerations in selecting from the pool of potential data sources. A series of six sections follows on common methods for collecting existing data, then three sections on methods for generating new data. Finally, five sections are presented on methods for processing and interpreting data.

Population Definition

The problem of population definition is essentially one of identifying the sources of potentially useful data. Identifying a data source includes simultaneous concern for the following three variables:

1. The context from which information or data is to be gained.
2. The content or class of information to be acquired.
3. The media by which existing information is stored and through which it will have to be extracted.

Figure 4-2 suggests the principal categories which serve to define these variables.

Sampling

Drawing rigorously defined probability samples through random or stratified random sampling (from which unbiased generalizations to the population of information sources are made) is not generally appropriate in system design. An exception is the generation of input to simulations, tests, and experiments involving system performance. That is, it is important to generate representative samples of information to be processed in system simulations and tests. In general, though, precise generalizations to the population are not required.

The following are more important considerations:

1. Don't overlook important categories of information. A useful technique is to keep an accumulative record of incoming

Contexts From Which Information Can be Gained

- ✓ Operating agencies faced with problems analogous to those to be resolved by the new system
  - Users and potential users of the new system
  - Agencies which interface with users and potential users
  - Sister services
  - Other nations
  - American and other industry
- ✓ Developers of similar systems and their components (including those involved in the development of the new system of concern)
- ✓ The general body of scientific-technical knowledge

Content of Information

- ✓ The information environment with which the new system must cope (for example, the flow and exploitation of military intelligence)
- ✓ Planning and historic
- ✓ Design
- ✓ Assessment and research
- ✓ Manpower and training
- ✓ Financial and economic
- ✓ Organizational (structures, missions, procedures)
- ✓ Facility
- ✓ Objectives, needs, and requirements
- ✓ Laws, policies, directives, regulations
- ✓ Socio-political, military expectations

Media of Storage and Dissemination

- ✓ Individuals
- ✓ Activities
- ✓ Objects (for example, existing equipment)
- ✓ Records
- ✓ Documents

Figure 4-2. Categories of Source Variables

information. If all categories are covered, then new information should not affect interpretation of the data.

2. Don't miss important single sources or classes of sources of information. Important single sources include public relations, financial, and point of approval data. In principal, ensuring the opportunity (via the sampling plan) to contribute to the data pool is nearly as important as an actual contribution from the source.

3. Separate fact from fiction. False notions of a user's current operations are very easily obtained. Two of the best ways to be led down this garden path are: to ask higher level managers about specifics and to analyze outdated/unused operating manuals.

4. Separate needs from wishes. An accurate understanding of potential users' needs is crucial to effective design. Surprisingly, one of the poorest ways to determine needs is to ask potential users what they want in a new system and to accept their statements uncritically. A more representative, valid sample of needs is likely from analysis of the potential users' actual use, or non-use, of information at the point of use.

5. Adapt data strategies to constraints. In most instances, time available for obtaining essential data is extremely short and late results are rarely useful. Therefore, the most practical strategy is to accept the best sampling plan within the time allowed rather than to insist upon a theoretically optimum sample which may exceed the limits. Furthermore, reliable estimates of the time required are difficult to make before the fact but can be more accurate, if based upon initial data gathering experience.

6. Account for bias. As already noted, practical data collection constraints (time, money, access to sources, etc.)

may preclude thorough sampling of the defined population. Thus, it is important and probably more effective to detect the nature of bias, its likely limits, and its implications than to assume the elegant sampling plan has probably controlled or eliminated biasing factors.

7. Know the limits. Knowledge of the entire range of variation for a given variable is not always necessary. Economics in data collection are often possible as a result and should be exploited. For example, if design specifications call for the system to perform without degradation only under maximum load, sampling system response behavior can probably be limited to variables which affect or generate maximum load information, ignoring all other cases.

Data Collection Methods

Six common data collection methods are discussed below. Each is treated from the standpoint of its strengths (likely uses) and limitations for acquiring information of significance to the system design effort.

Discussion and Interview

The principal means of communication for those involved in the design process is the telephone and face-to-face discussion or interview. These exchanges vary from casual to formal and from trivial to profound insofar as the course of development and final system design are concerned. Usually, no separate record of informal meetings among the design team members is made. The impact of these meetings upon current design work leaves an informal record. Separate records are prepared, however, for formal meetings between different management levels or between users and designers. Such records help to prevent future arguments about agreement terms and reduce wasted motion from backtracking over earlier discussions.

Occasionally, a broader review of information and opinion concerning an aspect of design is profitable. In that circumstance, carefully recorded discussions and interviews clarify the status of knowledge about the problem. It is almost always useful to have a set of questions thought out ahead of time, although it is also usually fruitful to encourage interviews to go beyond the scope of specific questions.

The most comparable information from different interviews is obtained by asking multiple choice or rating-type questions. However, seldom in data collection for system design is exclusive or major reliance placed on close-ended questioning. Extracting insight from knowledgeable persons is almost always more useful than conducting a scientific survey.

Incident Reporting

Factual information is best obtained from reports of specific identified incidents or events. Asking individuals to discuss their impressions of or reactions to the reported events is also useful. It is important, however, for the report to be as factual as possible. If incident reporting covers a broad range of subject matter, a relatively large number of incidents are required to provide a comprehensive picture of the situation. Discussions with many individuals are necessary, then, since most individuals can only reliably report a handful of incidents on a given topic. This means that incident reporting must usually be combined with other techniques if it is possible to contact only a limited number of individuals.

Document Review

Reference to a variety of documents for many specific purposes is an integral part of system design. Sometimes design documentation becomes extremely bulky and disorganized. Then, it is necessary to extract information and reorganize the documentation for efficient future design. Always be alert against the possibility of such documentation from documents becoming "busy work."

There are natural sequences from one type of manual to another in system development. In particular, design documents are used for assessment

planning; both design and assessment documents are used in the preparation of operation, maintenance, and training manuals.

Review of Records and Processes

It is almost always easier to use impressions or opinions of past system processes and performance than to actually dig up and analyze existing records and process descriptions. The review, however, yields substantially greater accuracy and insight. Since impressionistic recollections and records often disagree, a common assumption is that records are right and individual impressions are wrong. This assumption is not necessarily accurate. When there is serious discrepancy, it is worthwhile to reconcile the differences. Often, the very individuals who report distorted impressions of past processes and performance resolve discrepancies when summaries of old records are brought to their attention.

Observation of Operations

Many of the same considerations that apply to the review of records also apply to the observation of current operations. It is particularly important to compare observed operations with verbal and written descriptions of those operations.

For an initial orientation, observing existing operations in an unguided way is often desirable. However, recurrent or prolonged periods of observation are inefficient unless specific purposes, sampling procedures, and routines for the observation are established.

Survey or Questionnaire

While written surveys and questionnaires serve useful specific purposes, they represent a serious potential hazard. They look like an easy way to collect a lot of background data cheaply and quickly. However, a number of factors limit the apparent usefulness and advantages of questionnaires and surveys. If carefully designed, they take a long time to prepare; they should be pre-tested with a small representative sample; the time of respondents must be counted as part of the cost; there are usually a significant proportion of non-respondents; and busy operational and design personnel take a dim view of

surveys unless their purpose is immediately relevant and important. In all, the use of surveys and questionnaires for system design purposes should be selective and severely limited.

Data Generating Techniques

Frequently in system design, the data available are not sufficient to support required activities. Either the needed data are generated or assumptions about the data are used in design. Usually data having some apparent validity are generated, but which must be interpreted with caution.

Analysis of Trends, Projections, and Forecasts

Many of the parameters around which a system is built (for example, future input load and output demands) are not known at the time design decisions must be made. By developing credible models, obtaining relevant current and historic data, and extrapolating future conditions, data are generated for decision-making. Because of the substantial error margins inherent to most extrapolations, it is advisable to determine the regions of sensitivity and indifference of the design to extrapolated variables. Certainly, sound design demands a high degree of robust indifference to extract values of extrapolated variables.

Simulation

In the chapter on design assessment, simulation is defined in a precise and rather limited sense. Here, it implies efforts to represent and to exercise the system under design--and thus includes most assessment techniques. Simulation is used as a major technique to generate data which help guide design efforts. It is important, of course, to be sensitive to differences between simulation and operational conditions and the probable effect this will have on simulation results.

Rational Analysis

Hopefully, all of the analysis carried out by a design team is rational. Here, the term rational analysis means the process of breaking requirements or objectives into increasingly specific components. This is an extremely important part of the design process since it is the principal way in which functional, equipment, software, and human performance conceptualizations of the system are generated.

Data Processing Techniques

The variety of data processing techniques used in information system design are as broad as the entire field of data processing. They range from simple qualitative data collections to sophisticated multi-variate statistical treatments. This handbook does not summarize or review this field; many standard references are available. The comments below describe the relationships of data processing to problems associated with developmental data.

Data Reduction

The body of quantitative and qualitative data available and relevant to information system design is often chaotic and overwhelming. A difficult aspect is that many of the data will have significant utility only at the time they are collected. Other data are of recurrent use across a long time span and variety of purposes. Deciding ahead of time which data will and will not be of continuing use is very difficult or impossible. Thus, it is important from earliest design to hammer incoming data into a compact, consistent, and manipulable form.

Coding and Classification

The principal way of reducing incoming data to manipulable form is to categorize incoming data and code them on a real time basis. Even roughly pre-coded data are more useful than raw data that "someone should get around to doing something with sometime."

More elaborate classifications can be designed later if they are necessary or desirable. Early and informed attention to rough coding and classification procedures has a beneficial impact throughout the developmental history of the system.

Tabulation and Listing

The principal argument for getting messy qualitative data into machine-compatible form early is that low-level processing, tabulation, and listing are then feasible and facilitate the task of planning more powerful processing routines. If the data bank grows beyond non-trivial proportions, the time required for experimental tabulations and listings by hand exceeds available time resources. This does not mean, of course, that the whole bulk of data must be reduced to machine compatible form or go into an automated data bank. Rather, it means that some analog of each major item of data--extracted or coded--should be reduced to machine compatible form.

Description

The major role of data in system design is to produce analogs or descriptions of the system. These are in the form of requirements for system processes, estimates of system performance characteristics, etc. System descriptors should be initiated in the earliest planning of the data bank. Unless the data bank and procedures for its use facilitate easy description of the system along a variety of dimensions, considerable effort is being wasted in data gathering, storage, and retrieval.

Inference

When data processing mechanics are established, interpreting the data for design still remains. Mathematical or statistical models seldom apply to the kinds of data available. Allowances are often made for many biases and artifacts. As a system designer, you are essentially a creative artist and decision-maker in the face of ambiguity and uncertainty. Careful use of the best obtainable data prevents many false starts and erroneous conclusions. But, in the final analysis, data only provide a platform from which you can make the broad inferential leaps required to accomplish effective design.

CHAPTER 5

DESIGN ASSESSMENT

Assessment refers to any effort to determine the merit of system characteristics on rational grounds. It follows that assessment problems--from analysis of tentative design ideas to large-scale tests--pervade every stage of system design-development. This chapter describes the nature and role of assessment in that process.

Chief consideration is given to generally applicable concepts and techniques. In practice, however, these fundamentals must be augmented by applying experimental and analytical evaluative techniques specifically adapted to the circumstances at hand. A foundation for assessment efforts (as required by the design-development procedures detailed in Chapters 6-10 and Appendix 2) is examined under the following:

1. Assessment stages--typical phases of an effort.

2. Assessment interfaces--defining relationships.

3. Assessment factors--planning, conducting, and synthesizing an effort.

Two closely allied aspects of assessment are not included for discussion. First, the relationship and contribution of assessment to project management are dealt within Chapter 3, Organization and Management. Second, administrative test and evaluation procedures are not discussed at all, since these matters are specified by United States Air Force research and development policy in regulations and directives.

Assessment Stages

Although assessment efforts by nature vary tremendously in objectives, scope, etc., common stages of execution can be identified for all. Each stage contributes, materially, to the ultimate success or failure of an effort and should be included, particularly in those cases where a less formal approach is adopted. There are four stages:

1. Plan. A guide or blueprint is needed which informs those concerned about the assessment task. Typically, it covers the purpose, objectives, test methods or design (procedures, criteria, standards, data analysis, materials, and test equipment), support equipments (manpower and facilities), report intentions, and costs. Certainly, every effort does not demand a fully documented plan in the detail just listed. But, every plan should be formed on the basis of considering each point on the list. Furthermore, plans should be initiated as early as possible and refined as more accurate information becomes available or as decisions are made which affect implementation. Although a chore to prepare, there is virtually no substitute for an explicit, detailed plan; it will nearly always reduce controversy, wasted effort, and lost time while serving obvious positive ends.

2. Conduct. No matter how trivial, assessment efforts require data collection of some sort. The chief problems are related to the real differences between planning and doing. That is, one must assure that contingencies which invariably arise do not compromise the entire effort. Subtle, but important, changes in conditions and deviations from essential procedures are just two common error sources which may invalidate the obtained data for its intended use. Unfortunately, operational (field) settings which usually offer the most realistic conditions also present greater risks of compromise since establishing and maintaining necessary controls is more difficult. For this reason, in particular, it is desirable to pre-test data collection procedures, even when presumably trained and experienced personnel are involved in the assessment effort.

3. Synthesize. Reduction, analysis, and interpretation of the collected data and information is ordinarily the most demanding stage in time and effort. In fact, it may require as much time to complete as the other three stages combined. Ironically, how smoothly synthesis of results moves is almost entirely a function of adequate plans (Stage 1) and faithful execution of the plan (Stage 2).

   Synthesis is really little more than transposing data from the convenient form for collection to a convenient form for interpretation against those questions which caused the effort in the first place. However, the diverse skills and experience generally required for any assessment effort come to a focus at this stage--there is no place for self-styled "experts."

4. Report. A record of some kind should be generated which captures the essentials covered in the first three stages. As noted above, formal test and evaluation requirements specify detailed coverage and in this manner preclude failure to document what was done. The compelling problem is one of assuring that informal evaluations, comparisons, and related decisions are set down in a communicable recoverable form. Furthermore, it is most difficult to obtain an adequate record of assessment actions during the early stages of design-development. To do so requires a workable means and procedure for noting seemingly insignificant evaluative decisions such as, say, selecting this peripheral unit over that one. Establishing the record system and getting those involved to use it is the problem. Yet, these informal assessments are no less important than highly visible, organized efforts.

Assessment Interfaces

From time to time during system development, questions arise on the relationship (or interface) between assessment and three closely allied study approaches; namely, research, analysis, and optimization. This discussion on relationships among the latter is solely to provide a consistent frame of reference for describing assessment techniques later.

Assessment - Research

Figure 5-1 presents an idealized view of the connections between assessment and research. From this standpoint, basic research establishes a fund of knowledge concerning phenomena which may subsequently prove to have practical implications. Applied research selectively explores these potentially important findings and details their form and limits in the interest of attaining specified capability objectives. Checking the feasibility of these outcomes further under the constraints of a concrete application is the task of assessment. As such, practicability investigations may continue throughout the period of reduction-to-practice. Integration of a successful technique/device into a subsystem or a subsystem into a larger system usually generates new assessment requirements.

Summarizing, basic research (1) questions and tests existing knowledge, (2) fills gaps or extends the frontiers of knowledge, (3) raises critical new or unanswered questions. Applied research increases both the certitude and confidence with which one applies a domain of relationships having practical, immediate, or long-range utility. Assessment increases the confidence that a given means can accomplish specified goals, intermediate, or more general.

Assessment - Analysis

Analysis, i.e., mathematico-logical description and/or solution of problems, is an important ingredient of assessment. In fact, since analysis involves value judgments, the converse is also true to a minor extent. To avoid ambiguity of this sort, analyses are subsumed under assessment.

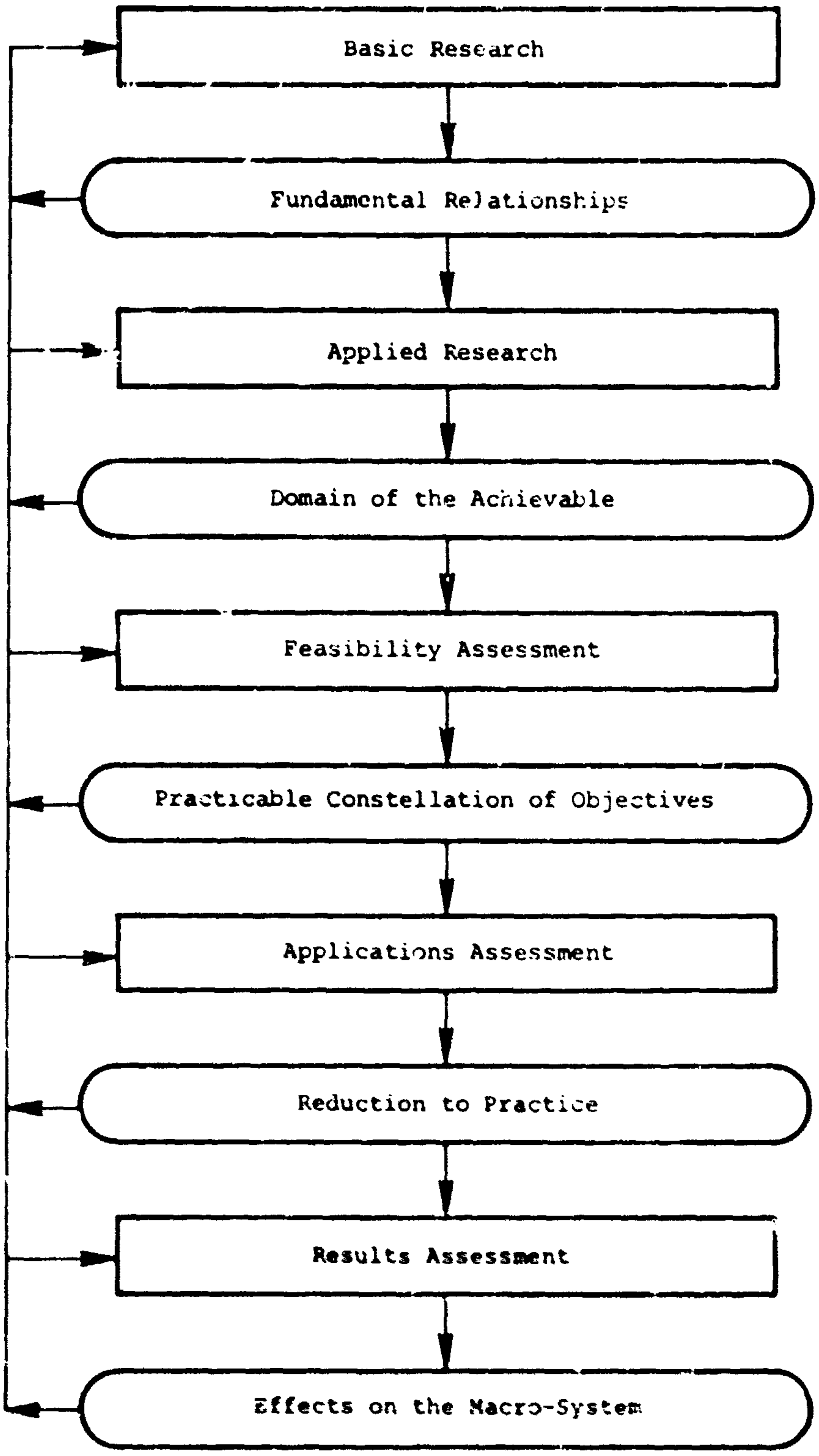


Figure 5-1. Relationships Between Assessment and Research

Assessment - Optimization

Assessment is frequently aimed at system optimization. It is directed, in other words, at determining how close actual performance approximates the most advantageous region of performance or a design feature fulfills overall design requirements. This concern occurs at two levels. First, there is the matter of optimizing input/output objectives and requirements as well as subsystem performance characteristics in relation to the entire system. Second, there is the matter of optimizing design characteristics to meet specific system objectives. In both instances, however, resolution is achieved through applying cost/effectiveness models to compare alternative objectives or designs and selecting that alternative which most nearly satisfies pre-established criteria. Thus, optimization can be said to be a special case of assessment.

Except in trivial cases, the number and complexity of alternatives makes an exhaustive study of all possibilities impractical. Consequently, there is a need for techniques which reduce the magnitude of data and computations required. There are many techniques for such purposes and new or refined methods are constantly developed for use. Table 5-1 lists several; others can be found by reference to Appendix 1. Unfortunately, assessment is hampered in these areas, due to a lack of methods for treating non-quantifiable characteristics in a rigorous manner. If such tools were available, evaluative efforts would make sense at an earlier design stage and contribute substantially more to steering a true developmental course.

Assessment Factors

Most, if not all, design assessments are dimensionable in terms of just a few key factors. It remains to identify these and consider their significance for planning, conducting, and synthesizing assessment actions.

Planning

As a matter of principle, requirements for assessment can and should be foreseen and integrated into the total system development plan. Effective

Table 5-1*

Partial List of Techniques for Optimization

I.	**Mathematical Techniques**
	Birth and death processes
	Calculus of finite differences
	Calculus of variations
	Gradient theory
	Numerical approximation methods
	Symbolic logic
	Theory of linear integrals
	Theory of maximum and minimum
II.	**Statistical Techniques**
	Bayesian analysis
	Decision theory
	Experimental design
	Information theory
	Method of steepest ascent
	Stochastic processes
III.	**Programming Techniques**
	Dynamic programming
	Linear programming
	Nonlinear programming
IV.	**Other Operations Research Techniques**
	Gaming theory
	Monte Carlo techniques
	Queuing theory
	Renewal theory
	Search theory
	Signal flow graphs
	Simulation
	Value theory

* Based on techniques suggested in ARINC Research Corporation, Guidebook for system analysis/cost effectiveness. Annapolis: Author, 1969. (AD 688154)

appraisal cannot occur in a vacuum, isolated from the main stream of effort, or as an afterthought. Recognize that such efforts draw from, and contribute to, the accumulative information base which constitutes the life line of the entire development process. Pragmatically, this means plans for such matters as subsystem/full system performance tests, critical hardware selection studies, and the like are formulated at the outset of the project or as soon as possible thereafter.

With this overriding principle in mind, we turn to other major considerations which enter into assessment planning. Of these, the first three establish a general orientation for the effort, while the second three treat specific determinations which are required.

Static-Dynamic. Assessment relies at one extreme upon paper-and-pencil analyses or upon exercising system components at the other. A choice affects, or is affected by, the nature of available data. Thus, a static evaluation utilizes facts in hand or those derivable from flow charts, machine drawings, mockup inspections, and related descriptive documentation to develop findings. Typically, these data are available during the initial stages of design and cost less to assemble.

Dynamic evaluation hinges on implementing the component functions, manipulating them under anticipated operating conditions, and collecting the findings of interest via systematic observation. Obviously, if carried out to the fullest extent, this approach necessitates that the component(s) in question (devices, software, and/or operating personnel) be available for such use--a circumstance which does not ordinarily occur until late in development. Additionally, a larger investment in time and effort is usually necessary to properly plan and conduct these assessments. There are also strong advantages: (1) validity--the extent to which observed findings represent the true situation--is more easily established; (2) less knowledge regarding the governing principal factors affecting operation, etc., is required; (3) results as well as experience gained are more or less directly applicable to ultimate operations.

Simulation offers an intermediate alternative to the opposing requirements of static and dynamic approaches. In this method critical component characteristics, functional relationships, and operating conditions are symbolically

represented; the representation is then manipulated according to principles understood or assumed to underlie the operation. Consequently, considerable knowledge of the governing principles and factors affecting operation must be available from existing theory, technology, or other empirical studies. Since component behavior is observed indirectly, the validity of findings is more difficult to establish and defend. Advantages of simulation are: (1) that component functions or larger system units can be studied well in advance of final design or a commitment to build; (2) alternative conditions, design features, operating techniques, etc., can be examined thoroughly at a relatively small cost in time/effort.

Experiment-Demonstrate. Assessment varies from carefully designed, closely controlled experimentation to open-ended, loosely controlled demonstration. The difference in orientation depends on the purpose. It may be to investigate cause-effect relationships systematically and enable confident prediction of results, or it may be to illustrate that certain results are attainable under specified conditions. In other words, an experimental approach is necessary to gain an understanding of component/system performance, while a demonstrational approach corroborates or presents those aspects that are already understood. Accordingly, experimentation is appropriate for selecting, refining, or modifying a specified aspect of design. Demonstration is suitable primarily for non-critical design reviews, capability determinations, or system exercises.

Intensive-Comprehensive. For any given assessment effort, there are practical limits on the amount of time and/or effort which can or should be allocated to its accomplishment. Basically, a balance must be struck between an intensive study of some narrow aspect of the problem and a comprehensive look at all aspects. For example, given a fixed, limited time to select among several candidate data processor configurations, one must choose to compare all candidates against a restricted set of selection factors; or, alternatively, compare two or three candidate configurations against a full set of selection factors including, perhaps, variations on a particular configuration, multiple test runs with varying input/output loads, etc. Due to the multiplicity of factors which enter into such a decision, there is no simple rule

or set of rules which determine an optimum solution. Rather, the implication is that in each individual case, planning must attend to an identification of critical determinants and effect acceptable tradeoffs.

Objectives, Criteria, and Standards. Assessment plans must set forth objectives clearly and unambiguously. To do so, however, entails distinguishing the two related concepts of criteria and standards. Figure 5-2 describes basic relationships among these concepts as well as their connections with the "macro-system," the external environment in which the subject system resides, and with assessment "measures," an important related consideration. Each of these three are discussed below, in greater detail, from a planning viewpoint.

1. Objectives. Assessment objectives identify the scope, detail, and precision of intended effort. As such, objectives:

   a. Represent commitments on the part of operational (user) personnel, systems engineers, and component specialists to priority goals.

   b. Provide a frame of reference for relating assessment requirements, i.e., the most cogent question put to any system-related proposal is: To what established objective(s) does it relate?

   c. Abstract essential information requirements at each level of system definition (circuit, component, subsystem, etc.) and indicate expected contributions or impacts between levels.

2. Criteria. Assessment criteria specify the basis on which achievement of a particular objective or set of objectives will be judged. In essence, these are stated, preferably direct measures of goal attainment or non-attainment. Assessment criteria are generally based on measures of the following:

   a. Timeliness--conformance with a specified upper, lower, or interval limit in time.

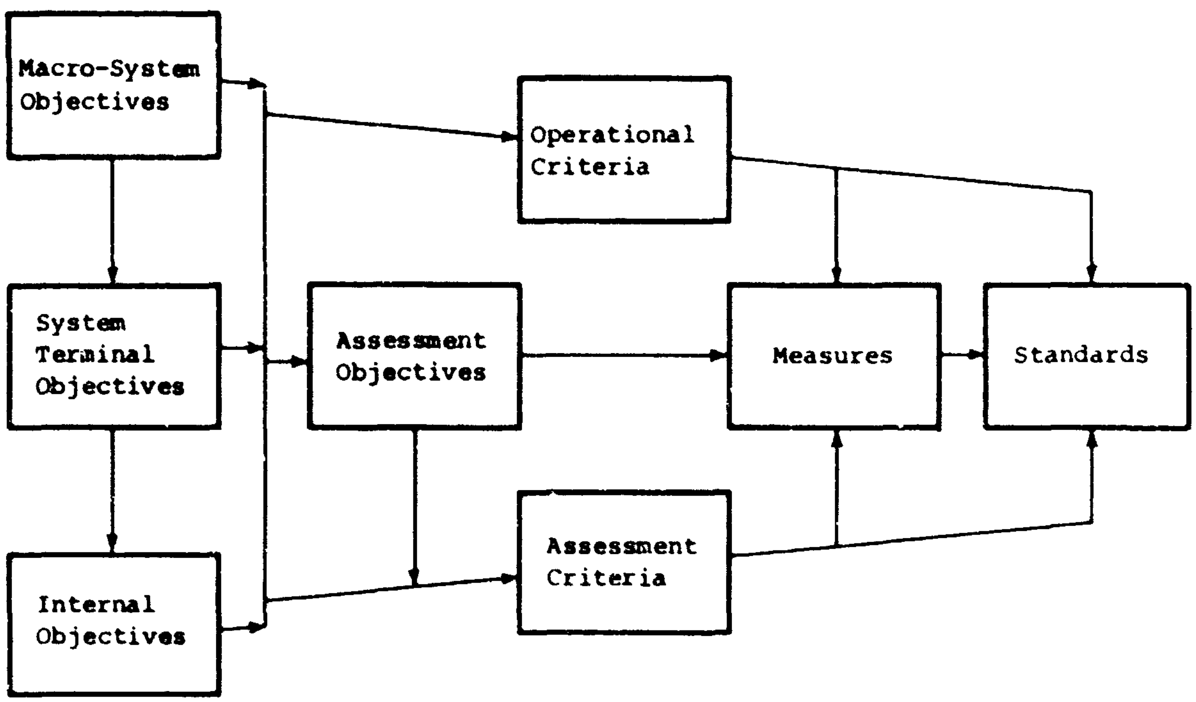


Figure 5-2. Relationships Among Objectives, Criteria, Measures, and Standards

b. Completeness--comformance with a specified range of coverage.

c. Efficiency--a ratio of input (aggregate costs) to output (useful work or benefit).

d. Sensitivity--degree of responsivity to change

e. Objectivity--degree of bias in a measure, result, finding, etc.

f. Validity--degree of correlation between a given measure and some independent measure; the degree of purity in scale, the lack of distortion, or the lack of contamination in a measure.

g. Reliability--degree of consistency in results with repeated measures.

3. Standards. Specific values of criterion measures established as a general gauge of adequacy in a component/system characteristic are referred to as standards. These measures can provide a baseline for judging system design and performance which extends beyond the confines of the immediate development effort. Of course, unique system-specfic standards may also be developed and employed as a part of the development process.

Measures. The selection and definition of assessment measures represents another major planning consideration. Useful guidelines for measure selection identification include:

1. Quantitative measures are preferred over qualitative; however, in many cases the only quantification possible is to count qualitatively defined units.

2. Direct measures of the variable in question are preferred over indirect wherever possible; direct measures are those which reflect change in the variable itself rather than a more or less closely associated variable to the one under study.

3. Measures are preferred which are relatively insensitive to changes other than those specifically designated for measurement; thus, measures which alter in significance as a function of any extraneous condition are to be avoided.

4. Measures which correlate with or are readily translated into operationally significant terms are preferred; i.e., where possible, one should select measures which are readily understood by user and management representatives.

Context. Perhaps the most difficult planning consideration involves defining a meaningful, realistic context in which to conduct the assessment. Three important aspects of context definition are described in the following:

1. Operational orientation. Primary attention should be given to the accurate inclusion of those facets of the operational environment which have implications for system development. In part this means that substantial assessment efforts must be devoted to checking the validity of operational assumptions, themselves. Since there is bound to be more explicit as well as implicit notions on "how things operate" than one can reasonably examine, selectivity among such assumptions is also paramount.

2. Anticipatory, generalizable results. It is essential to sequence assessments (and therefore contexts) such that results have maximum importance for subsequent design-development stages. The aim is to gain as much utility as possible from any assessment effort--and utility will generally be greatest when results obtained today assist in resolving tomorrow's problem. Unfortunately, this is easier said than done, since certain amount of clairvoyance and plain luck are necessary.

3. Assessment user orientation. The general requirements and specific intended uses of potential users for the results should be taken into account when the assessment is planned. For example, if the report generator routines are undergoing test and test results provided to the intended operational users for review, then it makes good sense to assure that the output format conforms to user expectations, whether or not format is a crucial test parameter.

Conducting

As noted earlier, conduct of the assessment effort is primarily a matter of faithfully executing a sound plan. It is rare, indeed, when execution is this straightforward. Frequently, the most carefully, thoroughly designed plan must be modified at the outset or during this assessment stage. When this occurs, constructive recovery actions should:

1. Seek reasonable assurance that constraints cannot be modified within established priorities for the assessment.

2. Document the probable (or actual) impact of a cutback on assessment objectives on design.

3. Abandon first those aspects and results of assessment least likely to yield significant design implications; strive to maintain those likely to have important implications as long as possible. When choosing among elements for retention and abandonment, it is important to recognize lost causes--that is, those elements which cause the effort to be impractical or too costly within existing constraints; to recognize the extent to which measures can be backed off from criteria, while still retaining essential utility; to base the selection on the best estimates of and appropriate weighting for:

   a. Expected degree of confidence in the relevant area of design before and after assessment.

b. Probable scope and impact of design changes likely to result from the assessment.

c. Net cost of assessing the particular element.

Clearly, one should not:

1. Waste energy attempting to remove constraints that cannot be eliminated.

2. Reduce or modify objectives too sharply, or abandon the effort entirely when constraints refuse to disappear.

Synthesizing

Synthesis concerns the analysis, reorganization, and interpretation of results to support the development of significant conclusions and recommendations--those which importantly change or maintain design-development direction. Generally, such information fits one or more of the following categories:

1. Identification of design faults, significant operational errors, or shakedown difficulties that can be overcome through redesign or operating procedures.

2. Distributions of performance under specified conditions that can support more precise normative expectations.

3. Improved and simplified models of the system (including input/output relationships) that can serve as a useful tool in operational planning to optimize utilization of the system.

4. Operational and maintenance strategies that maximize system effectiveness.

5. Requirements for personnel time and other system support.

SECTION III

SYSTEMS DESIGN PROCEDURES

This section details in five chapters the process by which a sound design and development effort yields a tangible, responsive operational information system. The capsule overview presented in Chapter 2 of Section I is drawn in fine-grain procedural form.

According to the orientation of these procedures, design and development is organized in three major divisions of effort, as follows:

Early Design (Chapter 6)--describes the formulation and definition of a viable system concept; procedures by which the concept is translated into function statements; and the allocation of functions to hardware, software, and personnel. The outcomes of early design are in the form of "paper-and-pencil" design specifications. The system components have not yet been engineered or matched to existing technology.

Design Engineering (Chapters 7, 8 and 9)--describes, in serial fashion that which is actually a parallel process, the detailed design of system components-hardware (Chapter 7), software (Chapter 8), and personnel (Chapter 9). The relationship of these subsystems to early design is illustrated in Figure III-1. Hardware, software, and personnel considerations begin in early design and break out as subsystems following the completion of functional allocation (tentative and/or dedicated).

System Transition (Chapter 10)--describes the considerations and procedures involved in transitioning the system from design engineering to installation and operation. This chapter deals with the nature of user developer/vendor negotiations, with user implementation problems, and with testing approaches oriented toward user evaluation of the system. System transition considerations emerge upon unification of the hardware, software, and personnel components for installation purposes.

The relationship of the five section chapters to one another and to the remainder of the handbook is illustrated in Figure III-1. The scope of the section is represented by the emphasized area.

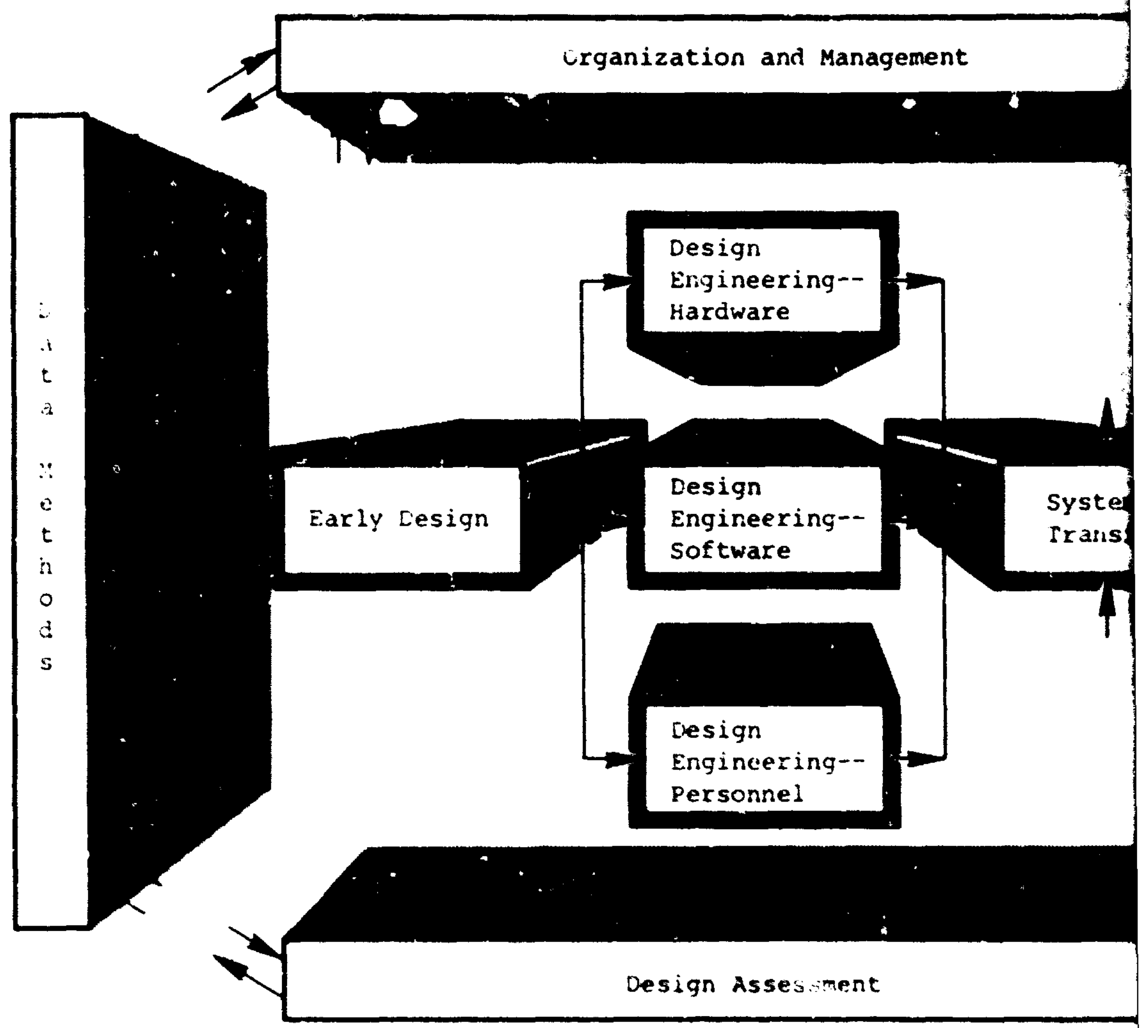
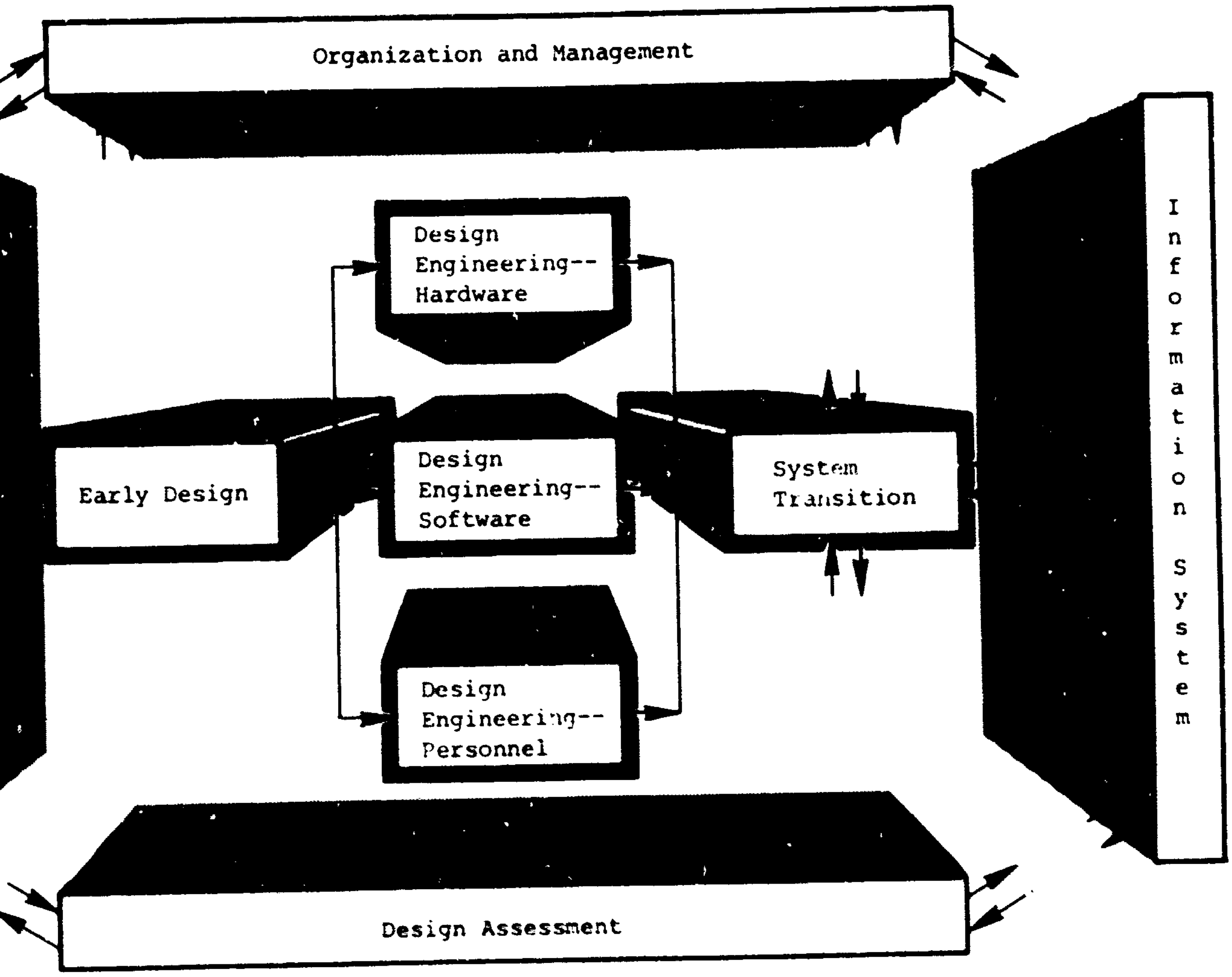


Figure III-1. System Design and Development



System Design and Development

CHAPTER 6

EARLY DESIGN

The process of designing information systems is a complex series of creative decisions integrated with data collection, assessment, and management activities. The efficiency of the design process and the effectiveness of the system are directly related to the nature, sequence, and accuracy of the decisions. These, in turn, are dependent upon complete and appropriate information and upon the procedures used. Thus, as a system designer, you need a useful and generalizable structure or set of procedures which guides the direction and order of design activities. This chapter outlines the requirements, procedures, and problems of early design.

Figure 6-1 is an overview of these early design procedures. The design activities identified in the overview correspond to sections of chapter content. Each major design stage and its component procedures are described in the same sequence within the chapter.

Figure 6-2 separates early design into the six major stages which terminate in functional specifications for a feasible design concept. The first derives a set of requirements and objectives for the system. The second stage defines the resources which can reasonably be brought to bear on development and operation of the system, as well as the constraints or boundaries within which such development and operation must occur.

The third and fourth design stages conceive the functions required of the system to transform available inputs into outputs. These outputs must satisfy stated objectives and the functions which lead to them must do so within established constraints. The third stage is concerned primarily with identifying and describing the system functions. The fourth has to do with allocating the functions to the hardware, software, or personnel components of the system.

The fifth design stage involves the integration of results to this point and provides the first major description of the system design concept. The sixth and final stage considered to be part of early design is concerned

with determining the feasibility of the design concept (in meeting operational requirements) and preparing a development plan for the design engineering.

6.1 Deriving System Requirements and Objectives (Stage I)

The derivation of system requirements and objectives involves the identification of one of two major kinds of "givens" that are imposed from the environment outside the system--namely, the givens or marginals for the output side. Input givens are discussed under the next design stage on resources and constraints. The two sections on functions design steps will deal more with "variables" or means within the system for obtaining outputs from given inputs. The sequence of major activities included in the process of deriving requirements and objectives is suggested in Figure 6-3.

6.1.A Defining System Development Boundaries

Defining the boundaries of system development is the initial focus of any systems effort. The types of information and the activities employed in selecting a development area and formulating a development rationale are shown in Figure 6-4.

Development Area. The thrust behind initiation of a design effort may have many sources. Operational problems with existing systems, economic squeezes, scientific and technological breakthroughs, acquisitions of new systems by competing agencies, projections of future capabilities of other nations, emerging demands, new services and new policy, directives, and regulations can play a significant role in motivating initiation of a system design. Many counter forces and cross currents influence the development and operational implementation of new systems. Moreover, conflicting motives can occur among superordinate levels of management, operating groups, the design team, and groups playing ancillary roles in design. Since there are usually valid reasons or needs for many new systems at once, competition for the economic and other resources occurs. Selection of an area for development, or at least elimination of some areas, must be based on a set of criteria which include the information system attributes described in Systems, Process, and Products.

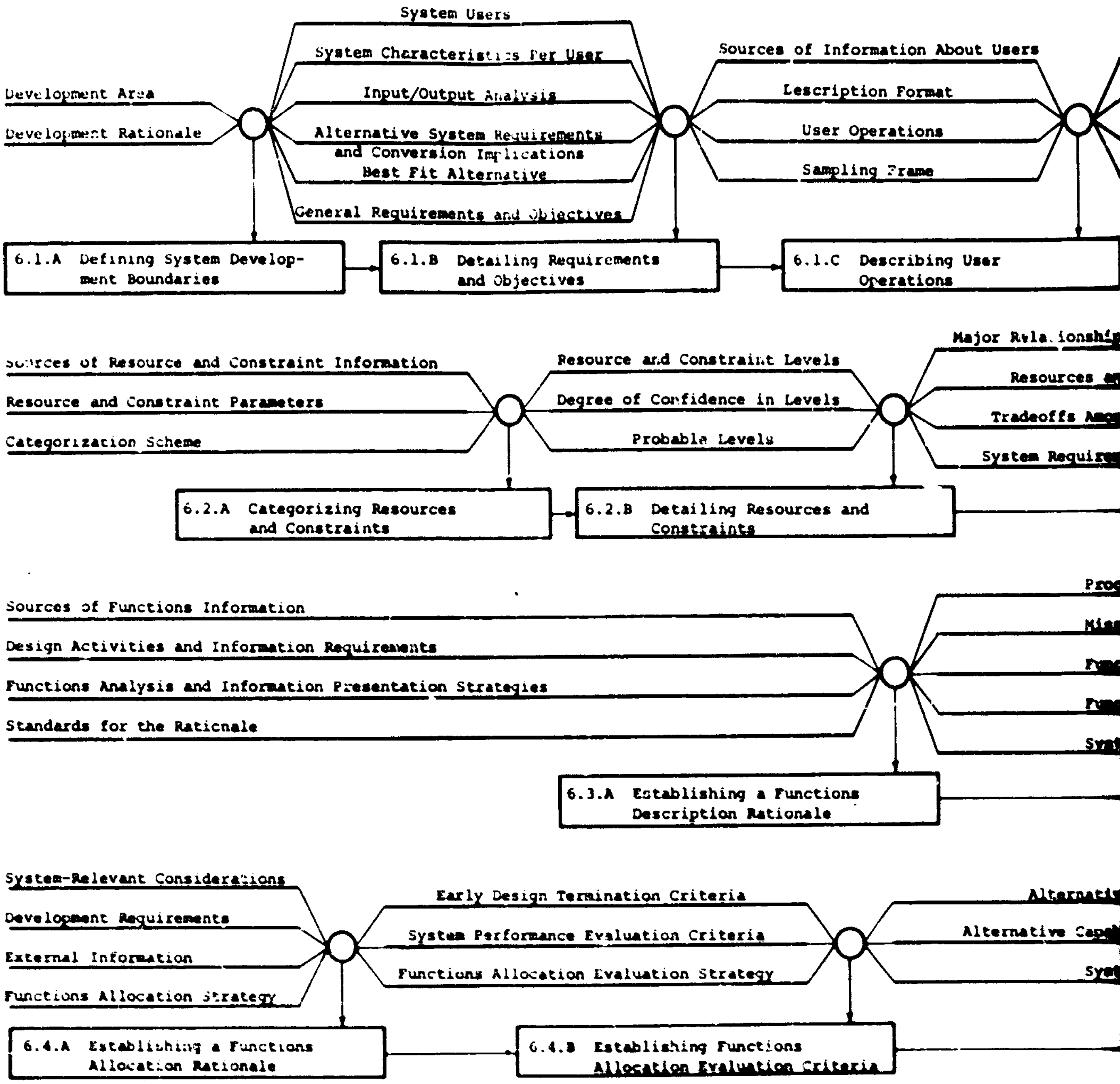
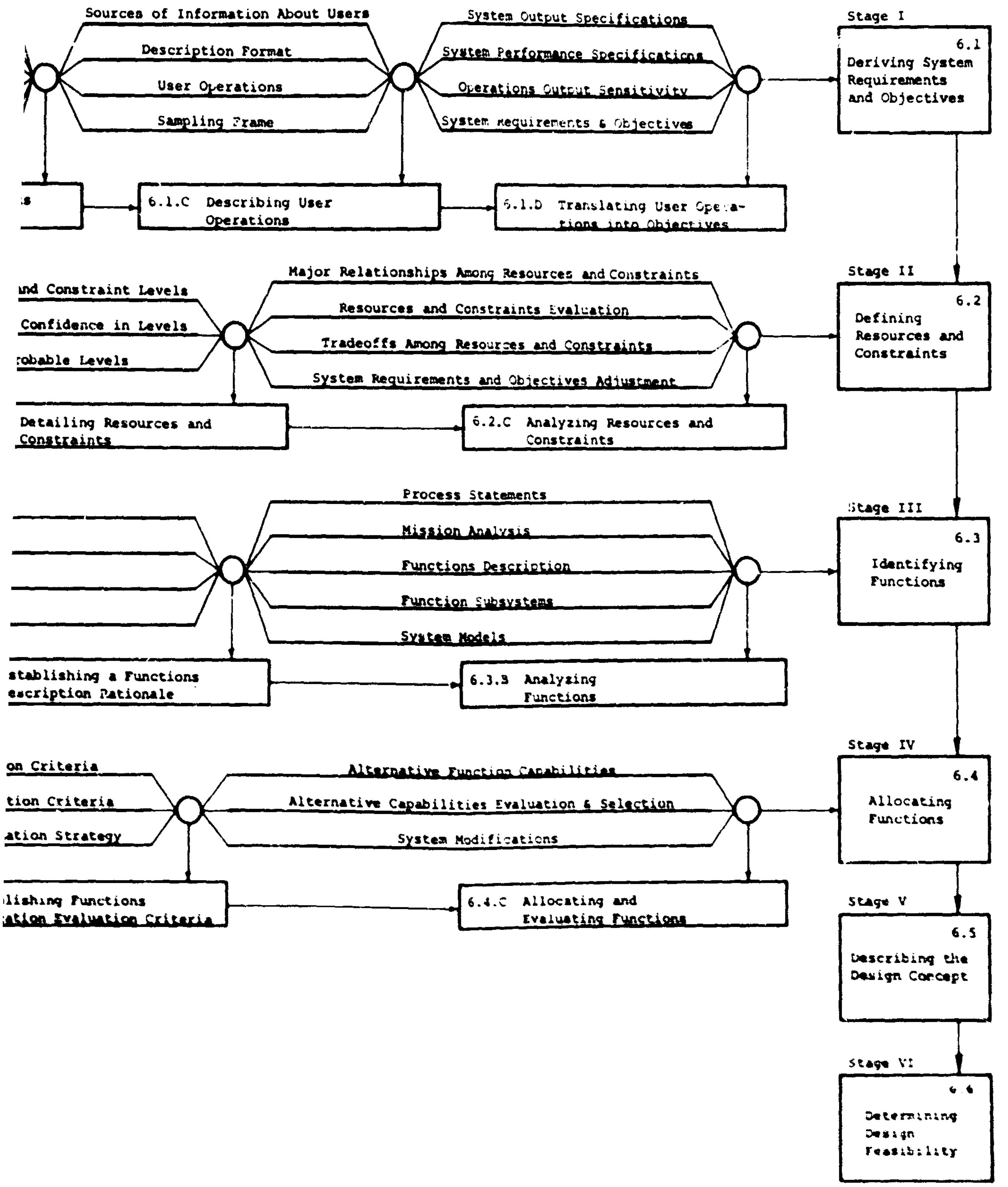


Figure 6-1. Overview of Early Design Procedures



gn Procedures

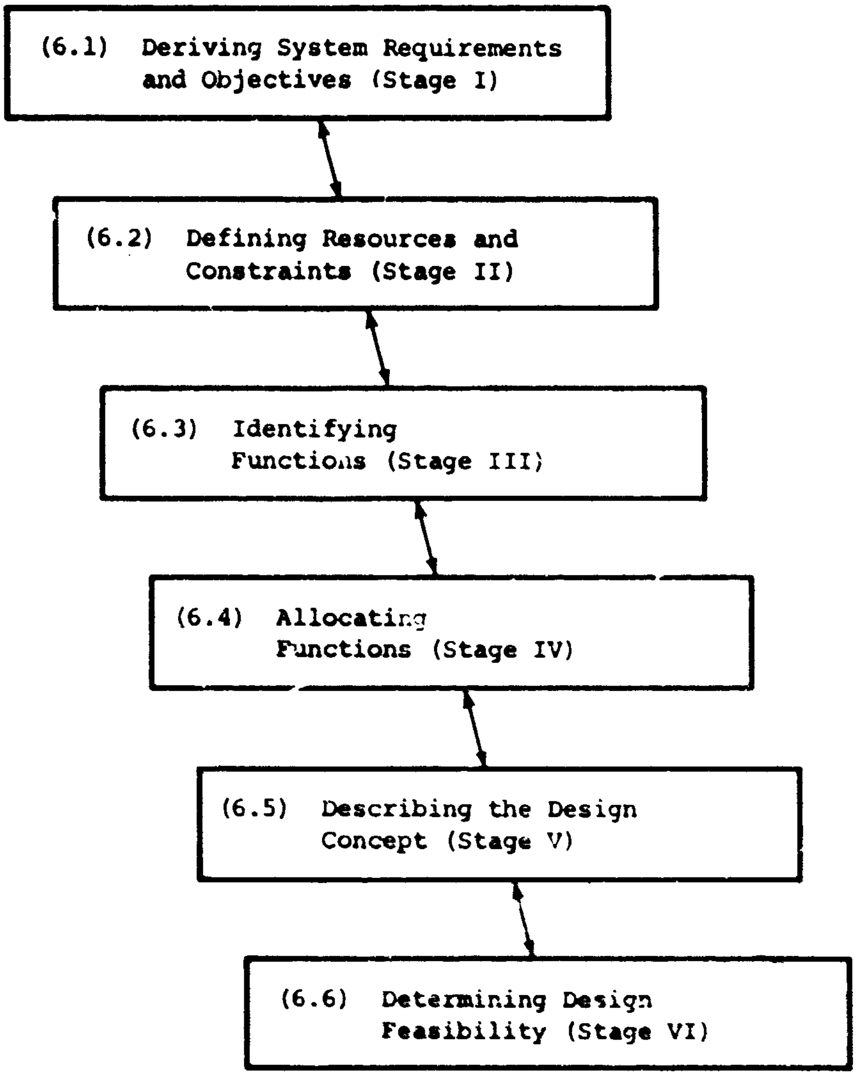


Figure 6-2. Stages of Early Design

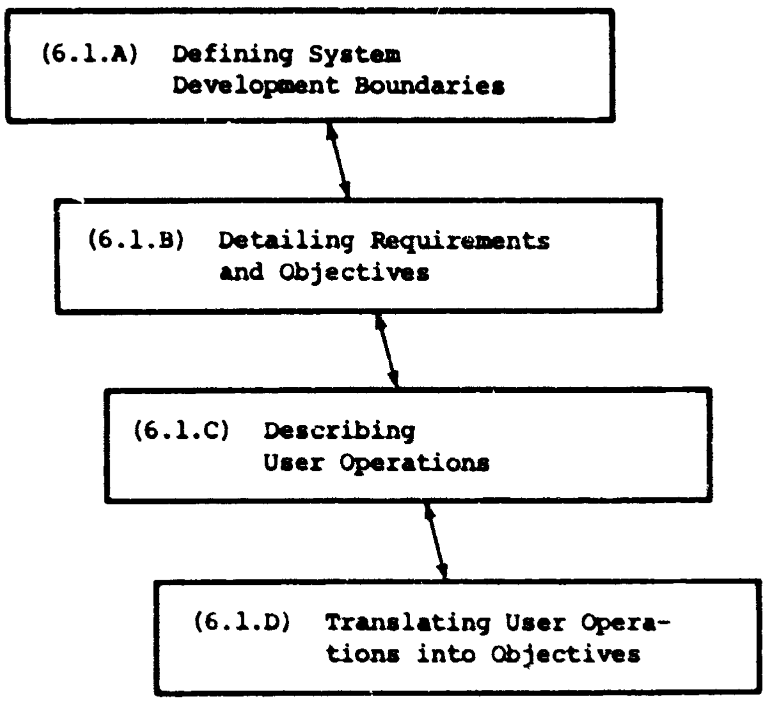


Figure 6-3. (6.1) Deriving System Requirements and Objectives (Stage I)

In order that selection of an area can be made, it is necessary that descriptions of possible alternative designs be gathered and presented in such a fashion that the established criteria can be applied. The information required in these descriptions includes:

1. Identification of the need--demonstration of the way in which present operations fall short of expectations or requirements, or the way in which they will be inadequate to meet future needs.

2. A description of how the contemplated effort would correct or eliminate the problem.

3. Feasibility of the solution within identifiable time and cost restrictions.

This information then becomes the starting point for the formation of objectives.

There will be instances where no clear-cut first choice or priority area evolves, and because of an inability to differentiate among priorities, a "command decision" is required. It may occur because of real equality between alternatives or as a result of unavoidable subjectivity in the evaluation. Instances of special interest and/or rigidity on the part of selected members of a design team are neither rare nor indicative of faulty procedure; they are inherent in the process. Having followed the procedures, however, the individual making such a "command decision" selects the development area, using all relevant information and being as objective as possible.

Development Rationale. An honest effort to identify the principal motives which stimulate the developmental effort and which will influence the course of that development helps to clarify the complex of communication required throughout the developmental process. Of particular importance is the nature of communication required between the design team and general management. Examination of these motivating forces contributes to the rationale for development of the system design. Additional areas which contribute to the definition of a design rationale include: implications of existing operations and resources and constraints, the extent or scope of the

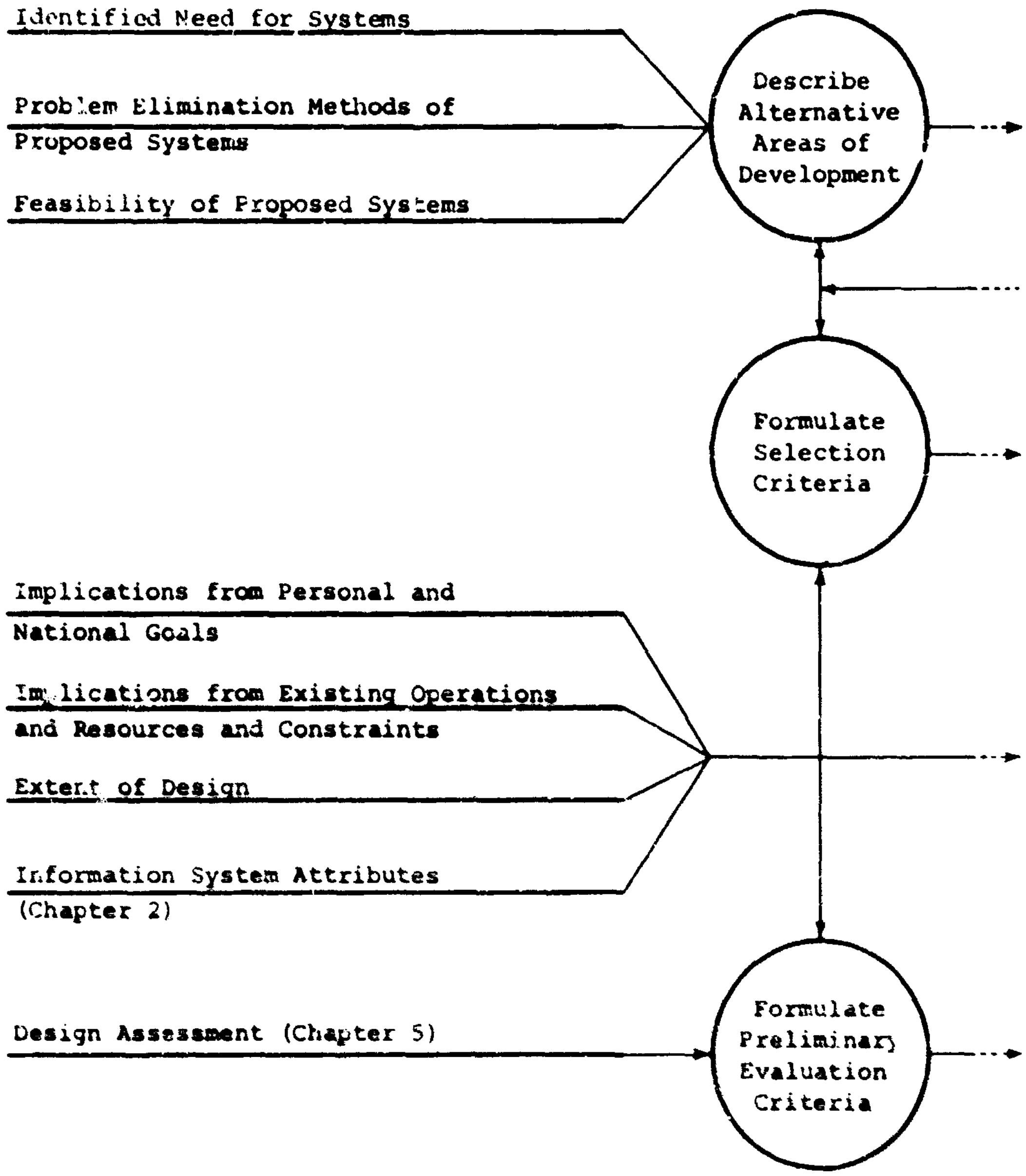


Figure 6-4. (6.1.A) Defining System Development Boundaries

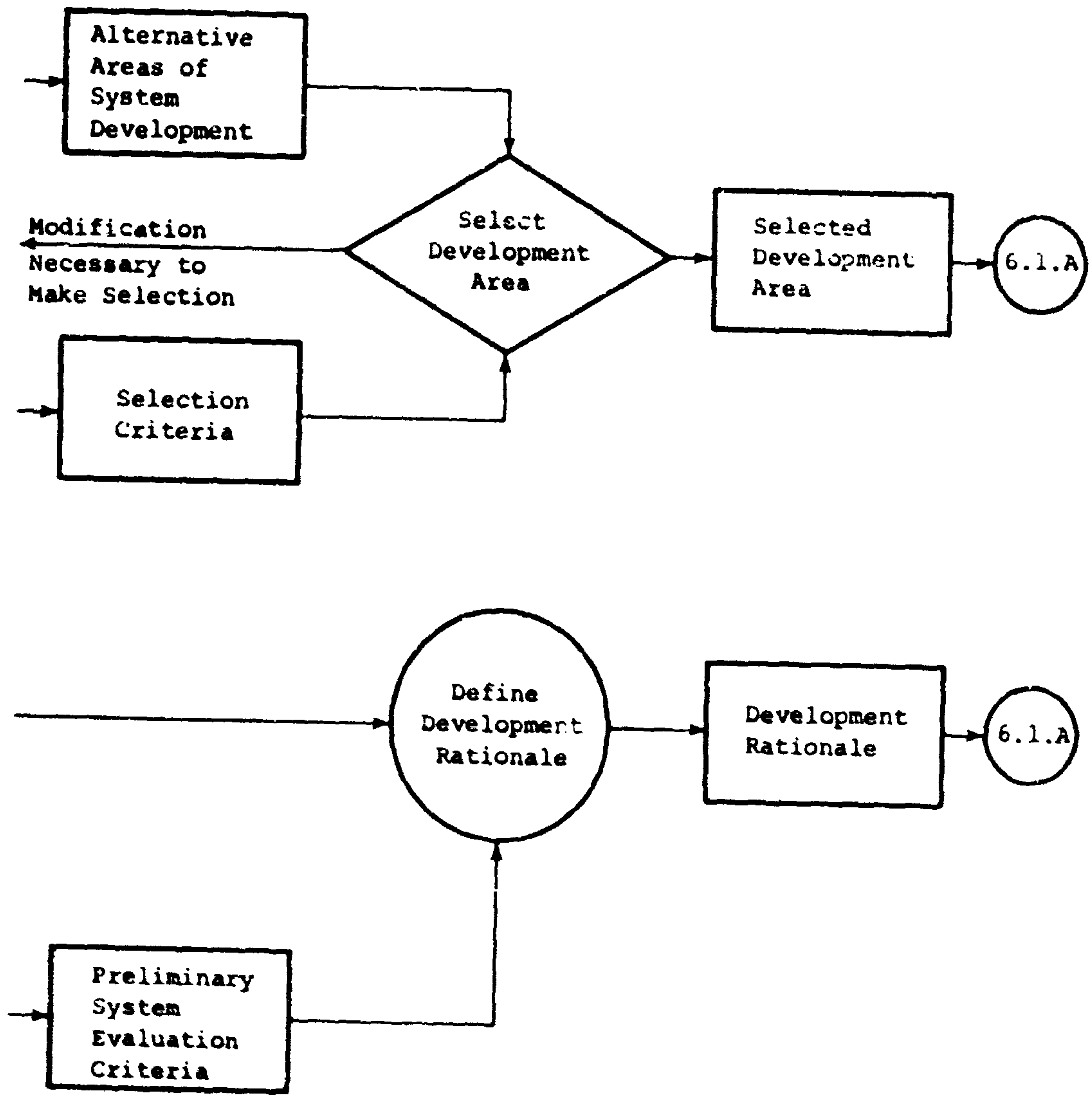


Figure 6-4. (Continued)

design effort to be undertaken, and the preliminary identification of criteria for successful design.

Implications from national goals. A prime determinant of areas in which the Nation is willing to invest research and development funds and time is the set of long-range objectives to which the Nation is dedicated. This information is often not available in any formal sense; however, a conscientious effort should be made to explicitly state any such goals that are, or could be, relevant to the developmental area. It is important to probe for, and be alert to, implications from national goals; it is equally important not to force implications or consider goals which are inappropriate to the development area.

Implications from personal goals. The natural tendency of management, regardless of size, is to make its operation the "biggest and best" of its kind. Management is not always aware of and sometimes not particularly interested in what is happening in other segments of the total operation, or the effect that change will have on other operations. On the other hand, management is aware that attached to larger and more complex operations are greater prestige and power--a greater sphere of influence. This is not a negative characteristic; rather it is desirable. However, the design team must recognize and temper the influence of this characteristic on system-design decisions. Since personal goals--of the user, of R&D management, etc.,--are the most fluid and least predictable source of input to the design process, implications drawn from them should be carefully weighed in terms of the potential life span of that influence in the system-development effort.

Implications from existing operations and resources and constraints. The problem of determining what the systems developers really have to work with is comprehensively treated in the next section. It is sufficient to say here that the implications derived from realistic resources and constraints generally fall into five major areas.

1. Equipment implications which commit already existing hardware or define particular hardware which must become the interface point with other systems.

2. Long-range schedules in which previous system development planning has mapped out courses of action. The schedules imply that particular capabilities, equipments, personnel, monies, etc., will be in specific configurations at certain points in time.

3. Fiscal commitments which relate to items 1 and 2 above and which act as a very real constraint on developmental activities.

4. Personnel availability in terms of the kinds of human capabilities which are or can be made available.

5. Areas of serious operating problems for which the expenditure of funds seems reasonable in light of their overall degrading effect on operations.

Extent of design. A further consideration in defining the boundaries of system development is determining the extent of the design area. There are three major levels which are considered at this point in development.

1. Machine programs. At this level the instructions for the hardware are prepared, tested, and taken to the field by the programmers--who then operate the system in whatever conversion mode is appropriate--until line personnel become sufficiently acquainted with the process and procedures to take over.

2. Programs and practices. At this level, the documentation of design is prepared and disseminated. This documentation describes both the machine process and the manual procedures in some fashion, but relies upon the ingenuity of each particular location of system operations to adapt the documentation for personnel training.

3. Programs, practices, and training materials. At this level of design, the documentation is completed and materials are prepared for training to support both conversion to and operation of the new system.

There are circumstances which justify each of these design levels. A design level decision must be made, at least tentatively, at this early stage of design so that the detailed planning for development strategies (1) focuses on the totality of future developmental problems, and (2) does not concern itself with the kinds of decisions which should not involve the design team.

Preliminary system evaluation criteria. From the earliest stages of design, it is highly desirable to begin identifying the criteria by which the effectiveness of the system can ultimately be judged. For example, even a preliminary definition of evaluation criteria facilitates and clarifies the delineation of system requirements and objectives. Detailed criteria appropriate to evaluation of the system emerge only on the basis of evolving design and detailed analysis. Information system attributes, described in Systems, Process, and Products, apply to the definition of preliminary criteria, as do many of the considerations in Design Assessment.

The development rationale should be carefully documented. It should describe the system and its known parameters, specify the organization's intent to develop a certain area, tentatively identify the types of products which will evolve from the design process, and provide preliminary criteria for evaluating the system. Part of this documentation, the statements of the analytical relationships which have led to the selection of the development area, are useful in providing an historical overview and are involved in numerous other design decisions as development proceeds.

6.1.3 Detailing Requirements and Objectives

Once a development area has been selected, a level of detail is required that defines the interfaces between the system to be developed and the world into which it must fit. The purpose of this activity is to analyze alternative

approaches to the specific development area so that the definition of "system" can be made and general requirements and objectives prepared to structure it. The activities and information employed in this process are shown in Figure 6-5.

System Users. A user in this context means a recipient of output from the system under design. Considerations up to this point have been with serving some unique set or sets of users, identified as "firm" users. There may be other users (or use situations) which are classed as "possible" users. That is, system design can be purposely configured so that it will or will not have an influence on their activities. Potential users and their use situations should be considered at this stage of design because of their utility in further definition of the system boundaries.

Each firm and possible user are individually examined to identify gross system characteristics which meet that particular user's needs. These are system characteristics which start to structure or set bounds on the system operation. Each user will suggest some of these characteristics. An examination of the nature of the specific output for each user defines possible system characteristics.

System Characteristics Per User. Detailed requirements and objectives are derived from an understanding of user activities which the system will support. Relevant user considerations include:

1. Identifying the types of users served in some significant way by the system, including unaided humans and humans assisted by equipment and other informational aids.

2. Defining the bracket of time over which the system will serve each type of user. For example, the design team cannot be content to look at the current information requirements of a given type of user, but should attempt to project his activities five or ten years into the future and derive from this projection what his information requirements will be.

3. Identifying locational and functional relationships among users which suggest that total system objectives

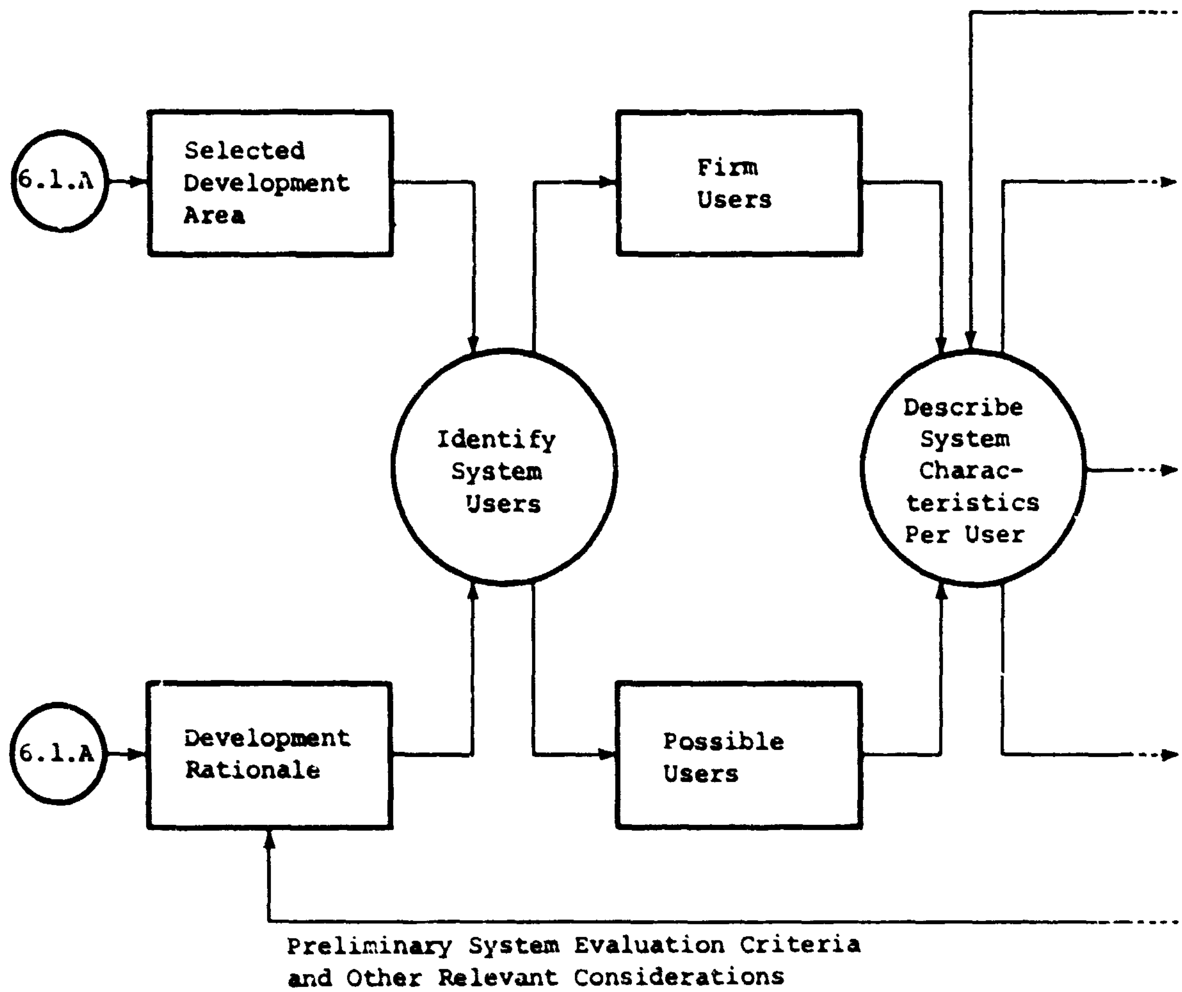


Figure 6-5. (6.1.B) Detailing Requirements and Objectives

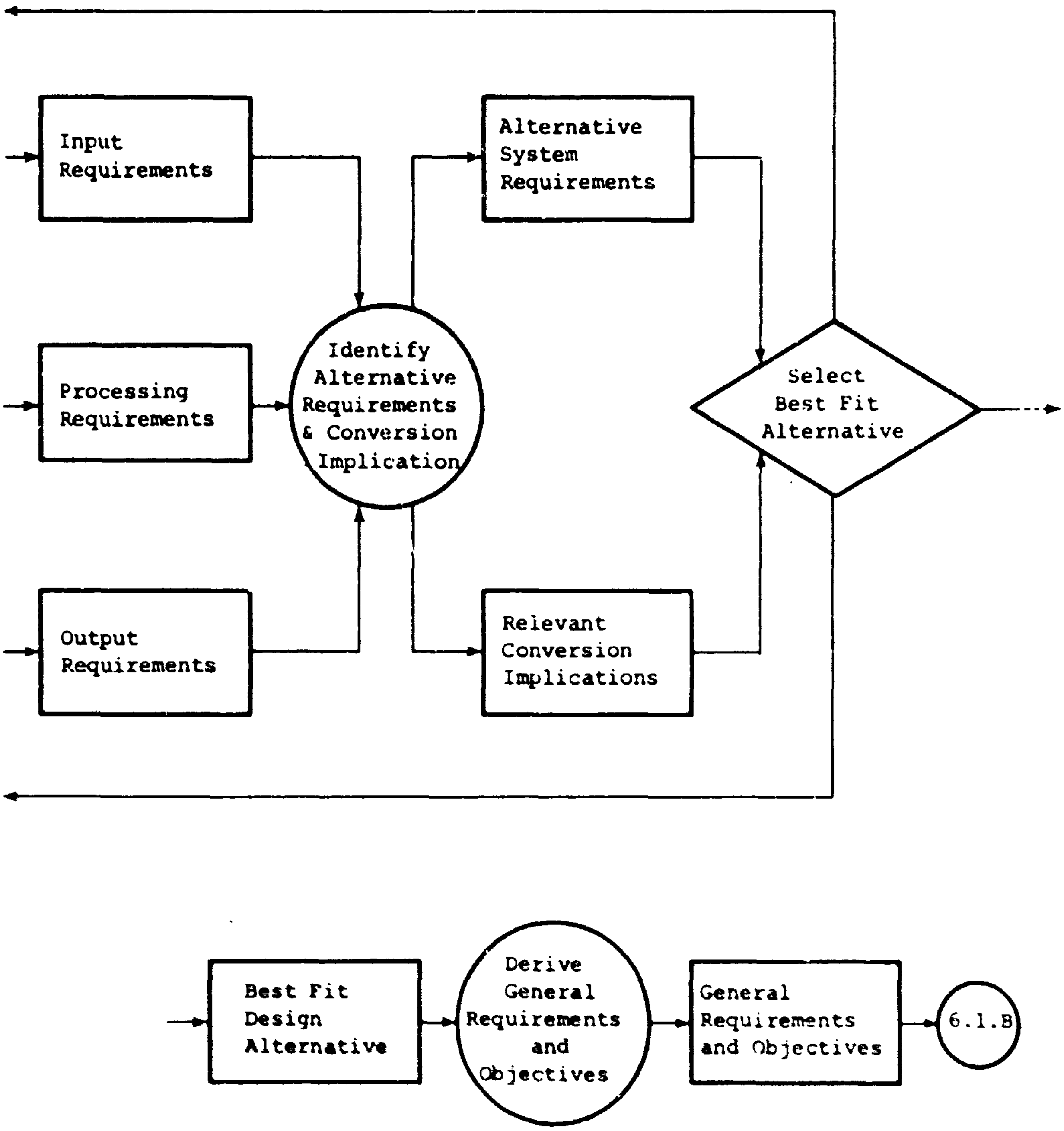


Figure 6-5. (Continued)

are different from the simple sum of requirements for different types of users.

4. Establishing boundaries relative to the needs of each type of user. For example, a specific user's stated needs might be considered outside the boundaries of this system design if his needs are better met by another system that will be concurrently available with this system.

Input/Output Analysis. Identifying users and formulating system characteristics and output requirements for each user is a creative process. That is, hypotheses are formed--and tested with available data--about how users will be served by any particular set of characteristics and outputs. Under each hypothesis about output requirements and system characteristics, a general analysis of the input requirements is performed. If the major functions of the system are already being performed and the main purpose of the development effort is to improve that performance, existing inputs which are appropriate to output requirements are easily identified. If the system is new in terms of its purpose and processes, existing sources of information provide inputs, but there will be many additional input requirements. With general user output requirements, identified along with associated inputs (those existing and those anticipated), it is possible to hypothesize processes which will convert those types of inputs to the required types of outputs. Each of these should be examined for each user, those to whom the system is dedicated and those who will benefit from its existence.

Alternative System Requirements and Conversion Implications. Based upon the preceding analyses, sets of system requirements which define alternative kinds of system operations are hypothesized. These prescribe alternative system perimeters, including and excluding users, as the combinations of system characteristics dictate. The alternatives should be made explicit, and their conversion implications should be associated with each requirement as with common groups of them. Particular kinds of systems necessitate or preclude specific kinds of conversions. The conversion process is often a unique systems development problem in itself. Its consideration at this point in early

systems design is critical, since conversion implications can control the selection of system users or system operating characteristics.

At this stage of systems development, descriptions resulting from these analyses will not be in minute detail. It is more important that all ingredients which contribute to the decision about what system will be designed are considered to the level of detail that the existing information permits.

"Best Fit" Alternative. Hypothesizing alternative sets of system requirements and associating operation and conversion implications with these sets utilizes all available information about the system. This information provides a basis for selecting the unique scope of the system to be designed. Most often, at this point in development, the "best fit" alternative is apparent. All of the design and operating implications are identified and examined in sufficient detail to determine the appropriateness of any particular set of system requirements. If the "best fit" is not evident, a review must be made of the total process of detailing the requirements and objectives to reassess all of the assumptions and hypotheses upon which it is based. If the review does not result in an apparent "best fit" alternative, the development area selection process should be examined.

General Requirements and Objectives. The formal output of detailing the selected development area is a set of general requirements and objectives which define the system. These requirements and objectives should be prepared very carefully and made available for review by management and operating personnel in the systems area which is affected by the design.

6.1.C Describing User Operations

Detailing the area of system requirements and objectives establishes a framework within which the relevant behavior of users is meaningfully analyzed and described. Analyzing user behavior includes the following steps.

1. Identify the best sources of information concerning user operations.

2. Further define the total population of users by characterizing each class according to relevant variables such

as availability for specialized training in using the system. Characterization is in terms of distribution tables, averages, variability, or limits.

3. Identify and delimit major contexts within which relevant user behavior takes place.

4. Identify major contingencies which the system will not be able to attend to and the reasons, as well as those contingencies with which it will be able to cope.

5. Identify the various kinds of missions carried out and objectives fulfilled by the user which are relevant to information from the intended system.

6. Break user missions into major segments, operations, and classes of suboperations.

7. Verify and refine the preliminary structure by trying it out against a small set of specific examples of user performance.

Figure 6-6 demonstrates how these activities and informations are employed in describing user operations.

Sources of Information About Users. At this stage of early design, the appropriate sources of information about system users and their activities are well defined. The sources will vary between systems and according to the extent of design being undertaken. In general, the sources can be categorized into:

1. Command management.

2. Line operating personnel.

3. System staff (quality control types).

4. Developers of other interfacing systems.

Description Format. Techniques to obtain data used in describing user operations are tailored to the particular user. availability of relevant information in written sources (e.g., research studies, job analyses, previous

system studies), access to users, and the contexts in which the users operate. Interviews and critical incidents make useful contributions. Task identification, description, and analysis however, are the most useful. A description format which portrays the specific operations of the selected users should be developed. An example of description parameters, sufficiently general to be applicable under most conditions of information system early design, are:

1. Users/using groups.
2. Functions or operations.
3. Tasks or activities.
4. Types and numbers of users.
5. Using conditions affecting user input requirements (user input requirements are actually system output requirements, of course). These using conditions are described by:
    a. Frequency.
    b. Volume.
    c. Use time.
    d. Perishability of information.
    e. Dependency of other operations on the results of an operation being analyzed.

Many of these classifications of information require definition within the specific system development context. The important thing is not that standard meanings be assigned to words such as operations, tasks, activities, etc., but that for each specific developmental effort, these labels are clearly defined and unambiguous

User Operations. In describing user operations, it is generally assumed that users require information to meet real needs. That is, users require information not just because they want it, but because their actions are influenced by the information in some way. Thus, a proper approach to identifying

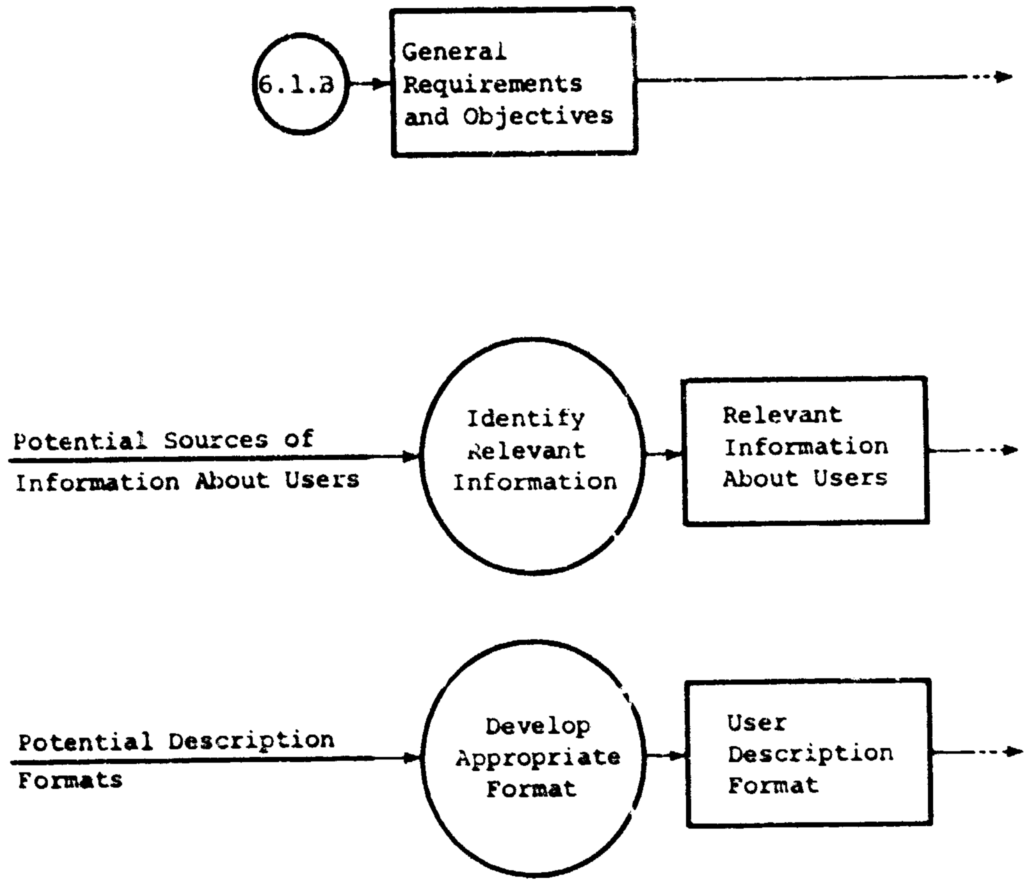


Figure 6-6. (6.1.C) Describing User Operations

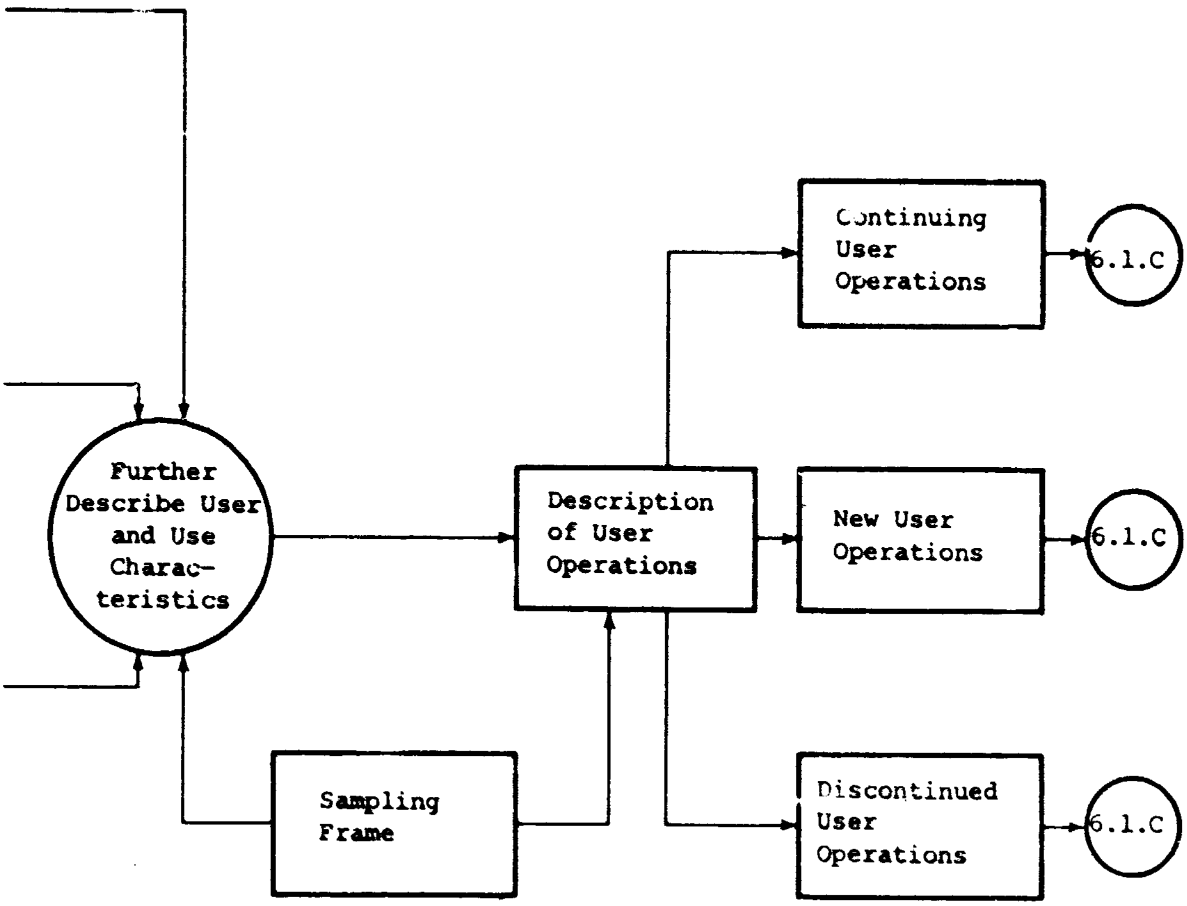


Figure 6-6. (Continued)

pertinent information requirements is to uncover the operations most sensitive to availability or nonavailability of potential system information.

For every user, the selected information category should be as completely documented as possible. Since user operations are often extensive, careful consideration of types of useful data contained within the format economizes time and funds required to describe user operations. Obviously, the information format requires much greater detail about user operations than just their functional modules. Detail, down to the task level, is required in some cases so that all operations are pinpointed and assessed.

In particular, three types of operations are desired:

1. Those operations which the user currently performs and which must continue to exist when the system goes into operation.
2. New operations which the user will perform as a result of the new system.
3. Current user operations which will be discontinued when the new system becomes operational, either because they will be performed by the new system or because they are no longer relevant to the operational flow.

Thus, out of this design activity comes the definitive description of the user under the new system and sufficient detail about those operations so that specifications for required system outputs can be set.

Sampling Frame. User operations may be too extensive and diverse for exhaustive analysis and description. Thus, it is necessary either to accept sampling or entirely abandon the derivation of information requirements from user operations. The problem in the past has frequently been that the sampling involved gross abstractions from the user domain, and the basis for the abstractions has not been clear. This results in gross bias in the definition of requirements.

Sampling means selecting some smaller set of user performances from a larger set of possible performances. The following points are relevant to sampling procedures:

1. The framework established by detailing system requirements and objectives serves as a direct contributor to the definition of a sampling frame.

2. The sampling bases and weights for choosing among different hierarchical levels, e.g., tasks versus specific behaviors, are often different. Tradeoffs between levels are also considered.

3. Sampling on bases other than frequency of occurrence is entirely valid. For example, presumed sensitivity to information can influence sampling ratios.

4. Random sampling here means enumerating all of the behavioral units among which sampling takes place. A primary motive is to avoid the exhaustive work involved in making such an enumeration. Thus, quota sampling is more appropriate than strict random sampling for the more detailed levels of user behavior.

6.1.D Translating User Operations Into Objectives

Implications for system design are derived from appropriate arrays of user [illegible]cations. The implications are directly relevant to performance outputs of the system, i.e., related to the system output characteristics required to support successful user performance. Figure 6-7 shows the activities required in the conversion of user operations information into a set of system requirements and objectives.

System Output Specifications. Estimates of output characteristics, derived from descriptions of user operations which must be supported, comprise the first set of system output specifications. The information types included under these specifications are:

1. The physical form of the output.

2. The information content of the output. (At this early stage, it is often possible to identify only types or classes of information rather than the details within any one class.)

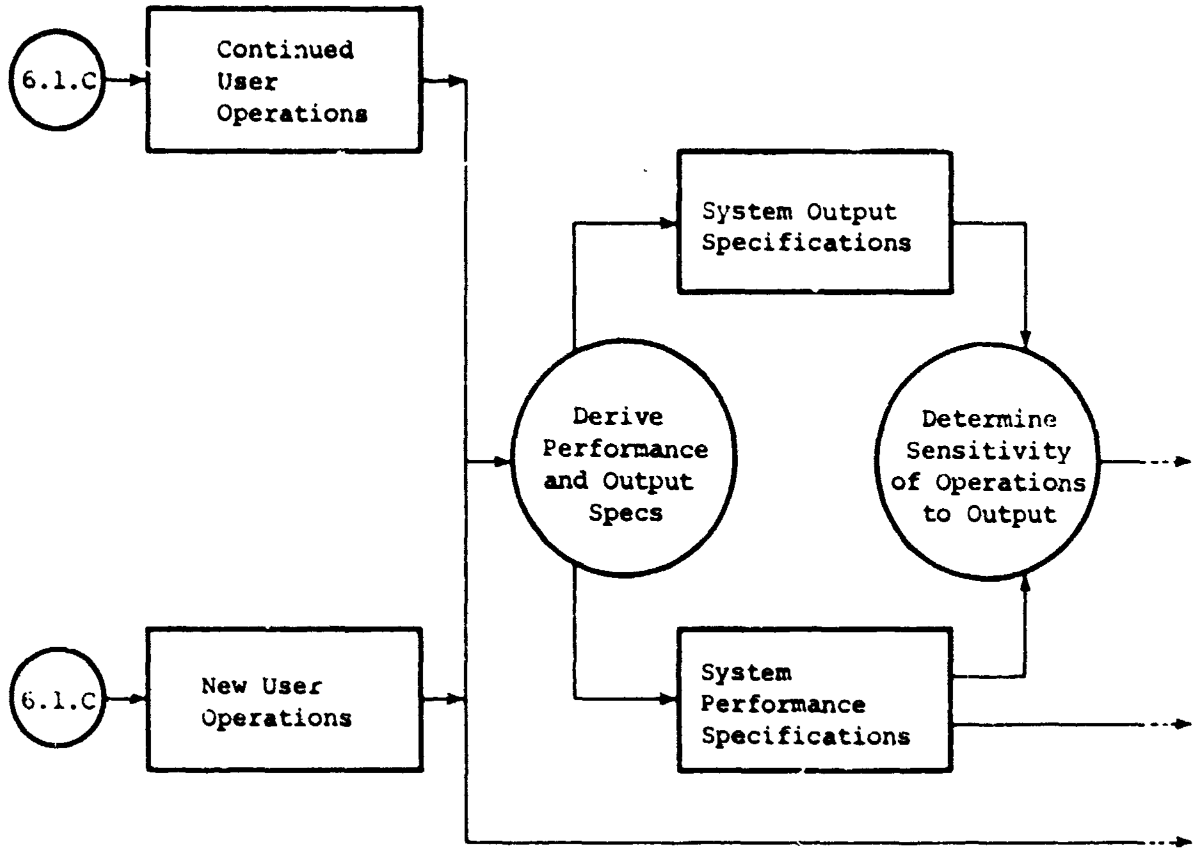


Figure 6-7. (6.1.D) Translating User Operations Into Objectives

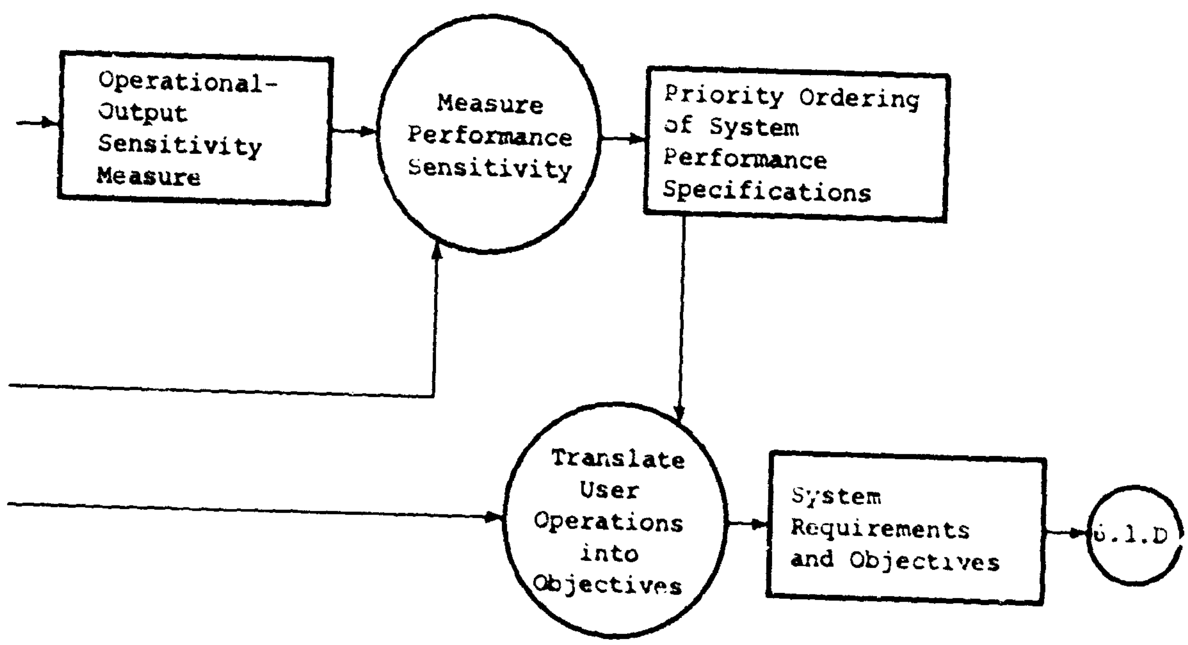


Figure 6-7. (Continued)

3. Frequency of output required.

4. Dimensions of accuracy to which the system output must adhere so that the desired reliability for user operations is attained.

5. Descriptions of output comprehensiveness when the information output to the user must completely describe the particular operation or activity.

6. Recency, which are statements of the effect of information currency on the user operation.

7. Reaction time, which is relevant to the reporting of certain kinds of aperiodic or nonscheduled events.

System Performance Specifications. Descriptions of user operations are also utilized to derive specifications of system performance and operational activities which produce desired output characteristics. The performance specifications are examined in terms of the potential range of information available to the system. It is then possible to determine which system operations have the greatest information requirements in order to achieve the necessary system output.

Operational-Output Sensitivity. All of the system output characteristics and specifications previously identified are weighed against the possibilities of achieving them. It is necessary to develop some type of priority measure, a measure of the sensitivity which operational activities have to output characteristics. Sensitivity can be measured by the difference between performance that will be achieved without any system information flow and the performance achieved with the best information imaginable. The degree of performance decrement as a result of partial cutbacks from ideal information indicates stringency of system information requirements The consequences of performance decrements for achieving user objectives determines the importance of the information. Consideration of these factors aids in setting appropriate objectives for terminal output performance of the system.

System Requirements and Objectives. Once the priorities of the system performance specifications are established, a set of system requirements

and objectives can be prepared which specifies the goals of system operations. The system and user operations descriptions are translated into system requirements and objectives in terms of:

1. Output.
2. Operating modes (at least for those areas dictated by resources and constraints).
3. Areas and groups of personnel skills required.
4. Operating costs.
5. Volume of production.
6. Frequencies of output.
7. Spatial locations for installation and operation of system components.
8. Communications requirements within and between systems.
9. Storage and security of system data.
10. Back-up systems or alternative operating modes.

Useful system requirements and objectives have the following characteristics:

1. They unambiguously communicate to management and members of the development team what the intended outputs of the system will be.
2. They facilitate measurement or assessment of the extent to which outputs are realized.
3. They are formed at a level of specificity which permits evaluation of the system's capability to achieve the objectives.

Once individual requirements and objectives have been identified, they should be organized into a logical and nonredundant structure.

6.2 Defining Resources and Constraints (Stage II)

Whether an element is considered a resource or constraint is largely dependent upon one's point of view. The limits on any resource can be conceived as constraints, and the region within any constraint can be conceived as a resource. No system development has unlimited resources of manpower, money, materials, facilities, or time. Each resource has additional organizational, technological, operational staff, policy, and administrative staff limits. No information system is entirely independent of the context in which it will operate. These factors should be identified and defined early in system development to preclude any incompatibility with the realities that influence its success or failure.

Limits, once identified, do not necessarily remain forever fixed. Tradeoffs are sometimes made. Ongoing development can suggest that resources thought to be adequate are no longer sufficient; that constraints thought to be acceptable are intolerable; or contexts thought to be ideal are inferior. None of these possibilities changes the basic desirability of organizing and analyzing resource and constraint information early in design.

Although shifts in identifiable resources and constraints are expected to occur throughout the developmental process and the operational life of the system, it is possible to structure the basic considerations which apply to the definition of resources and constraints in early system design. The principal activities involved in defining resources and constraints to the system and its development are shown in Figure 6-8.

6.2.A Categorizing Resources and Constraints

Categorizing resources and constraints involves identifying and examining potential sources of information and deriving from them a preliminary set of resources and constraints. This preliminary set is then categorized by system-relevant parameters which permit a more detailed description of system resources and constraints. The activities and information involved in categorizing resources and constraints are illustrated in Figure 6-9.

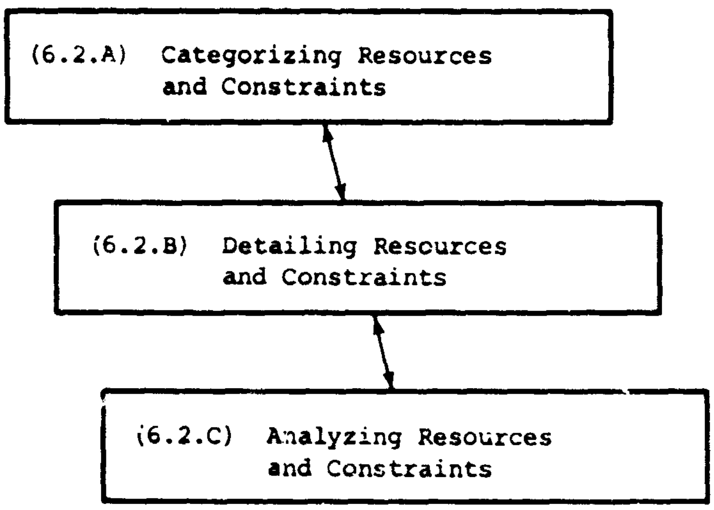


Figure 6-8. (6.2) Defining Resources and Constraints (Stage II)

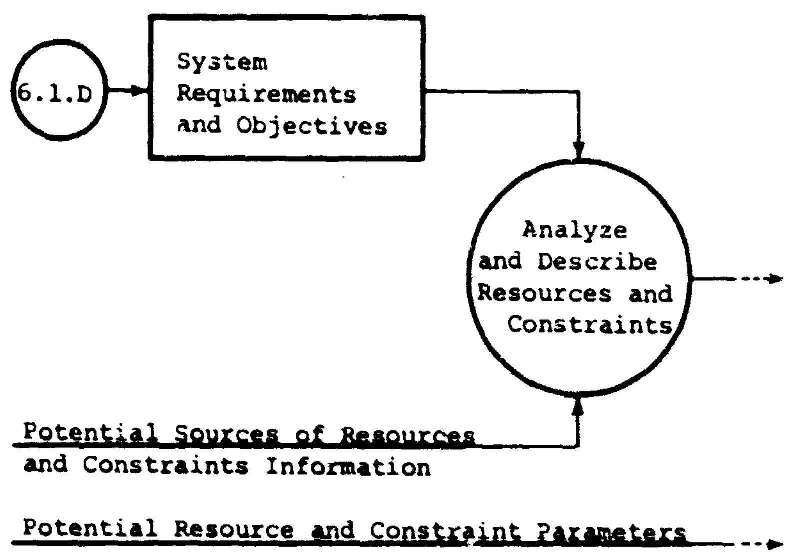


Figure 6-9. (6.2.A) Categorizing Resources and Constraints

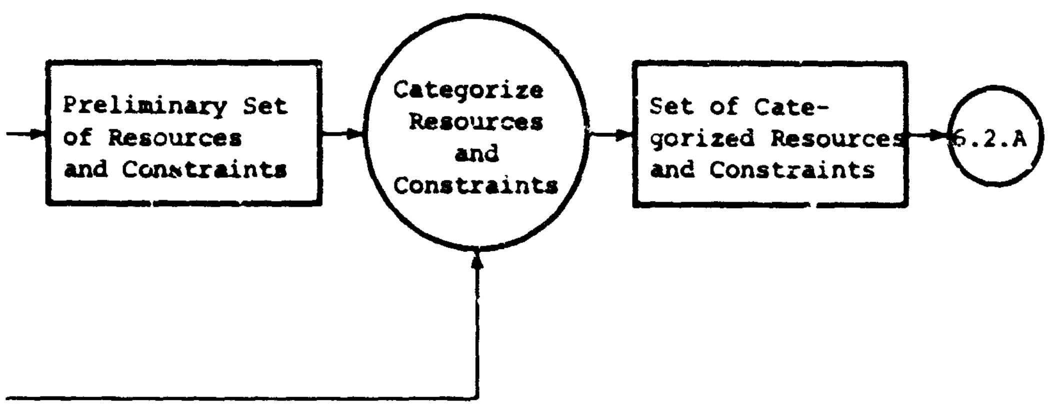


Figure 6-9. (Continued)

Sources of Resources and Constraints Information. Sources of information concerning resources and constraints are related in number and complexity to the dimensions of the system development objectives. The more encompassing the system, the more resources it requires, and the more constraints it needs to recognize. Some parameters are always examined--money, people, time--and some parameters remain peculiar to a given objective.

Potentially relevant areas of information for determining resources and constraints are described in the paragraphs which follow. Their interactive implications are manifold; the nature and extent of the interactions are specific to the given design objectives and conditions. Some resources and constraints information relate primarily to the final design and operation of the system; others relate primarily to the development effort.

Previous systems. Previous systems and their documentation are a particularly rich resource if the present developmental work aims to optimize or emulate the ongoing system. However, if the development of a new concept addressed to existing problems is attempted, resources and constraints information about the existing system is as misleading as helpful. Unless it is assumed that the previous system designers completed a thorough analysis of parameters and that nothing about the environment or the system objectives h-- changed significantly, a retesting of previous resources and constraints information is required.

Related studies. Related feasibility and design studies and their documentation--if available and applicable--provide sound resources and constraints information. Care must be exercised to insure that the relationship of the studies to the system under development is direct and appropriate. Further, it is necessary to be critical of the studies to assure that the study or documentation is correct and comprehensive, and that the information is acceptable.

State-of-the-art. The state-of-the-art in relevant areas of development is assessed for its potential contribution to the resources and constraints. The validity of these efforts to the development

program must be assured. It is unrealistic to base development on resources unproven by state-of-the-art studies or laboratory demonstration.

Management policy. Management policy or goals, at national and individual agency levels provide guidelines of resources and constraints. To interpret them literally is sensible; to ignore them is disastrous. The ideal approach to management goals is to derive guidance from them and attempt to effect change in them when necessary. Management policy can be interpreted as either a strong set of constraints and deterrents or a sound measure of resources and motivation. Furthermore, trends are probably as important as fixed policy. Most agencies operate under a complex of specific regulations which must be taken into account in the design.

Operational organization. Organizational structures are an important source of constraint information since it is difficult or impossible to create an information system which does not impact on the organizational structure. The extent of constraints imposed by organizational structures that interface with the proposed system must be determined. That is, the amount and kind of change imposed by the system is assessed against amount and kind of change the organizational structure will tolerate.

Traditions. Traditions, like organizational structure, are difficult to change. Traditions are sometimes as real and as constraining as regulations or contractual obligations. They are rarely spelled out in any document or set of references and are elusive and difficult to ascertain. Nevertheless, traditions cannot be ignored. A conscientious effort should be made to assess system objectives in the light of known or implied traditions. "Old timers" serve as a source of information about traditions.

Long-range plans. The long-range plans of the user agency and its ruperior organizations should be carefully considered since they determine the future direction of the organizations. If the proposed system compliments the plans, little more than pointing to

this consistency is necessary. However, if the proposed development effort does not coincide with long-range plans, reconsideration or change is needed. Rarely have all the long-range plans for an organization been pulled together in one document. Interviews with planning personnel and top managers augment, verify, and clarify planning documents.

Financial reports. Practically all information systems are justified, at least partially, on the basis of cost considerations. Accurate information concerning the costs of existing systems which are replaced provides a useful baseline against which justification for the new system can be formed.

Operational manpower. Early consideration is given to the personnel requirements for operating and managing the new system. Determination of the exact number of types of people, i.e., the knowledges and skills required to operate the system, like many other factors, is not possible early in a developmental process. Nevertheless, certain early assumptions are made, based on the best information available, about what is required. Job evaluations and descriptions provide a status report of skills and knowledges. If it is apparent that the requisite resources are not presently available and cannot be developed before the system is ready to become operational, a large problem exists. If, through selection, training, or job aids, the required knowledges and skills to operate the system can be developed in the required length of time, a resource is counted rather than a constraint. Again, early determination of manpower requirements is adjusted, refined, changed, and quantified as the developmental process progresses.

Developmental manpower. The considerations given to operational manpower resources and constraints are also given to developmental manpower. It is difficult, if not impossible, to definitize these manpower requirements early, yet assessment of developmental manpower is a critical requirement. Some assumptions, the soundest

estimates possible at the time, must be made. As development progresses and the requirements become more definitive, these assumptions are reexamined, clarified, and quantified repeatedly.

Facilities. Existing facilities are another strong area of resource, and sometimes constraint, information. Facilities include such diverse things as existing data base, existing and unused communications lines, computer time, building, and various other equipment or unused personnel capabilities. Historically, many system-development efforts have been motivated by the fact that one or more of these facilities was operating at less than capacity. However, improving the capacity of available facilities can be unnecessarily restraining, particularly if the development effort can profit from a new facility resource within the financial capability of the agency.

Time frame. An accurate fix on the time frame for development is needed very early in the developmental process. Information to make a definitive estimate of time required is not usually available early in the development process, but a reasonable estimate can be made on the basis of such resource information as time studies, feasibility analyses, past experience with similar development efforts, etc. This estimate is considered tentative and flexible and is adjusted as new information is assimilated during the developmental process.

User adjustments. It is reasonable to assume that people can adjust to only a given amount of change in a certain amount of time. Therefore, the amount of change which can be successfully introduced and assimilated by the personnel who will operate the system has a restraining influence on the development effort. The envisioned new system must be objectively compared to the present system to determine how much change is implied and what areas the change will affect. Two questions arise in regard to system changes: whether or not it is reasonable to expect the personnel involved to accept the change, and by what means a resisted change

is made acceptable. Areas such as multi-media training technology and techniques of system introduction are assumed in these considerations and planned for in the development.

System interface. System interface data are perhaps the most difficult kind of resource and constraint information to obtain. In order to acquire correct information, some initial assumptions about the boundaries of the system are first made. Next, each boundary is scrutinized for interface, interchange, or interdependent relationships with other systems. It is important to identify potential interfaces as well as real interfaces and to seek out the elusive ones as well as the obvious. Historically, failure to perform adequate analysis of the interface, interchange, or interdependent relationships with other systems has contributed to the downfall of many system efforts. Further, this step cannot be done once and then forgotten; it is repeated as more detail becomes available.

Personalities. Personalities exert a very strong resource-constraint influence on the developmental process. Complete systems have been built on the strength and enthusiasm of a single, highly motivated individual. Conversely, system efforts are sometimes retarded or stopped because of the attitude of one or two influential individuals. The opinions, intentions, and expectations of those individuals who bear the ultimate responsibility for system effectiveness and operating success have to be properly appraised. It is essential that these responsible personnel be kept informed of developmental progress so that their influence enhances the developmental effort.

By-products of the system objectives. Inferences from the system definition or the stated requirements and objectives contribute to the definition of resources and constraints. In particular, clarification of system input givens is often possible. The extent to which system objectives are explictly stated affects the extent to which input information resources and constraints are identified.

While every attempt should be made to collect this type of information, it is essential that implications not be overdrawn and result in unnecessarily restrictive conditions. Information should be collected in as many parameters as possible, without attaching unwarranted dimensions to those parameters.

Resource and Constraint Parameters. As noted earlier, some parameters of resources and constraints, e.g., money, people, time, are always examined. Yet, it is apparent that the potential diversity of sources and types of information creates such a mixture of resources and constraints that conversion to a "type categorization" is necessary. Although most of the vital information is boiled down to major categories--time and money--such an overdistillation prohibits careful analysis. Listed below are parameters of resources and constraints which, in most situations, permit a categorization scheme of common denominators. While some of these categories are irrelevant to particular situations, other situations require the use of additional categories.

1. Input information.
2. Time.
3. Cost.
4. Personnel.
5. Hardware.
6. Software.
7. Job performance aids.
8. Training.
9. Organization.
10. Facilities.
11. Laws.
12. Regulations.
13. Contracts and agreements.

14. Procedures.

15. Existing knowledge concerning the performance characteristics of similar systems.

Categorization Scheme. In categorizing information, it is important to avoid (1) drawing a categorization scheme so broad--i.e., too few categories--that it destroys the opportunity for analysis of interactive effects, or (2) drawing a categorization scheme so narrow--i.e., too many categories--that no transformation of the original resources and constraints can be made. Because of the necessity for evaluating the available resources and the constraining influences against the system objectives, selection categorization types should facilitate this comparison.

Since a source-type of information does not necessarily fit into only one categorization, each source of information must be carefully analyzed for its informational categories. For example, documentation from previous systems, a single source of information, can offer data for many categories of information. Consequently, each segment or item of information must be carefully analyzed for its contribution to each category of information. Overlapping the same informational item in more than one category is undesirable. That is, while an information source often feeds more than one category of information, a single item of information should not appear in more than one class unless it is suitably cross-indexed. Schemes for aggregating resource and constraint information must account for items given multiple categorizations.

Categorization of source information into categories is an arbitrary process. However, distilling the varieties of information to provide a more homogeneous set of information is appropriate and serves as a check on the reasonableness and structure of results obtained.

6.2.B Detailing Resources and Constraints

Detailed description of resources and constraints information is necessary before it has utility in the system development effort. A critical aspect of this description is determining the levels expected in resource and

constraint parameters. Admittedly, quantification is difficult and arbitrary. Yet, there is a potentially large number of direct and derivative quantification factors which are relevant in detailing resources and constraints. The following are generally applicable; their interactive effects are shown in Figure 6-10.

Resource and Constraint Levels. The range of levels which is expected in the system resources and constraints should be examined in terms of:

1. The probable maximum that can be obtained even with great effort.
2. The probable minimum that can be expected under the worst circumstances likely to occur.
3. The most probable level of resource is likely to achieve, based on the identified maximum and minimum levels.

Degree of Confidence in Levels. The degree of confidence in estimations of probable resource and constraint levels should be determined. This requires an analysis of the assumptions used in establishing maximum, minimum, and probable levels, and of the number of unknowns in the system to this point.

Probable Levels. It is important to examine the ease with which given resources and constraints can be modified. This requires identification of the criticality/noncriticality of the resource and the probable level likely to be achieved in conjunction with the extremes judged possible. Large discrepancies between maximum and minimum, coupled with a low level of confidence, indicate areas where the effects of modifying resources and constraints should be considered.

6.2.C Analyzing Resources and Constraints

The analysis of resources and constraints examines their interaction in the system. Analysis activities include analysis of resource and constraint relationships, evaluating the impact of individual and related parameters, and investigating tradeoff possibilities among the parameters. The sequence of resources and constraints analysis activities is depicted in Figure 6-11.

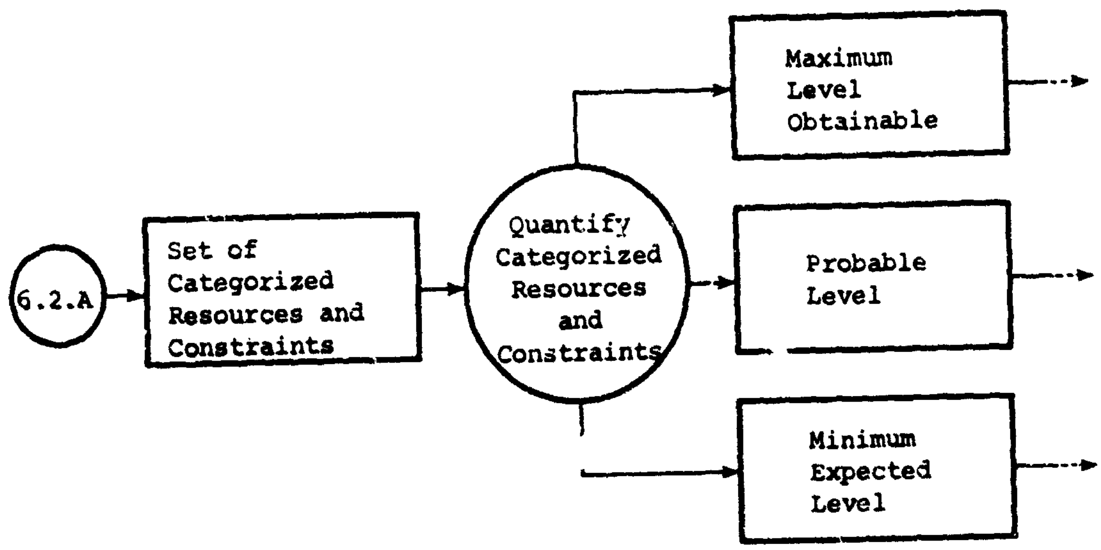


Figure 6-10. (6.2.B) Detailing Resources and Constraints

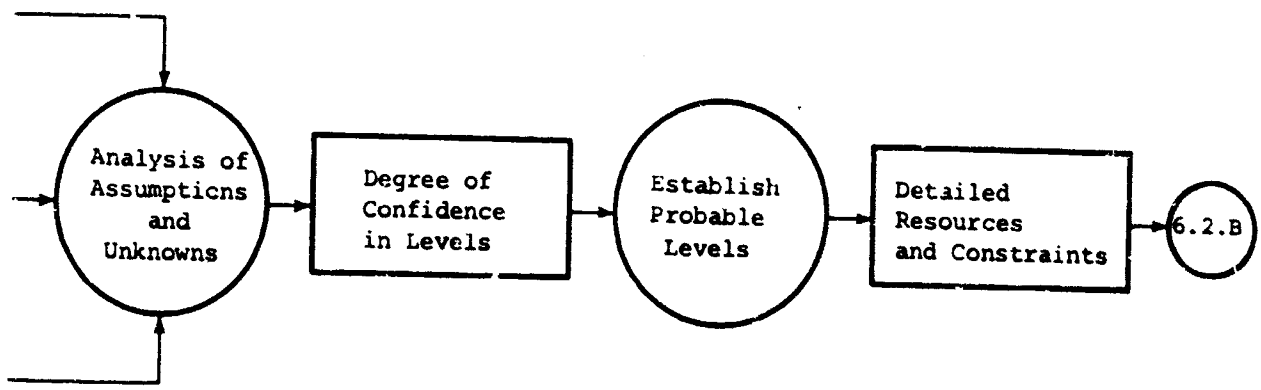


Figure 6-10. (Continued)

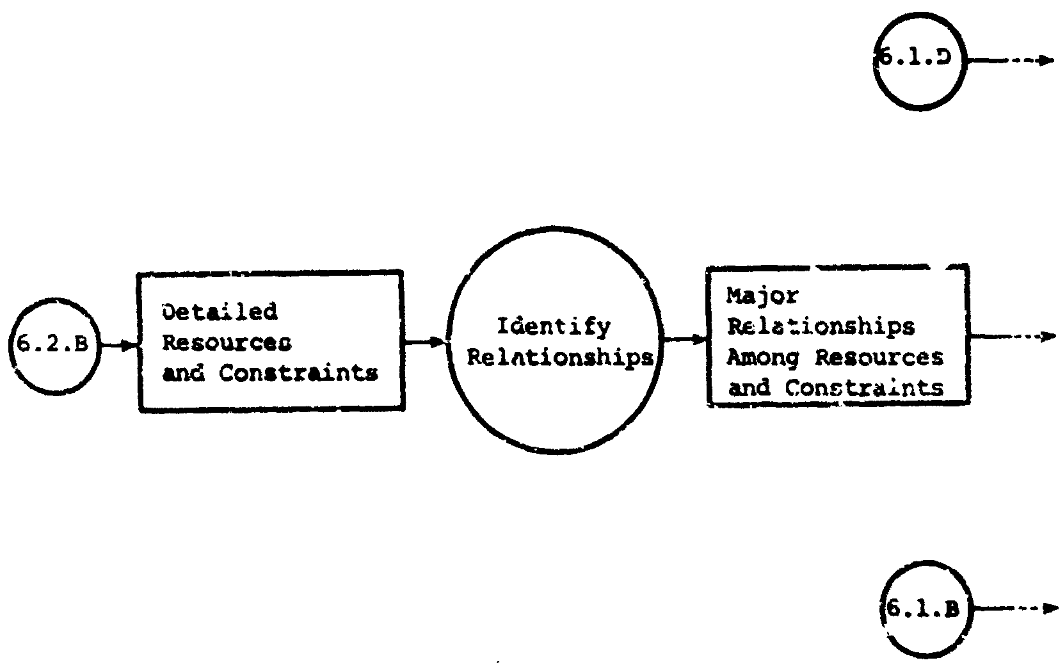


Figure 6-11. (6.2.C) Analyzing Resources and Constraints

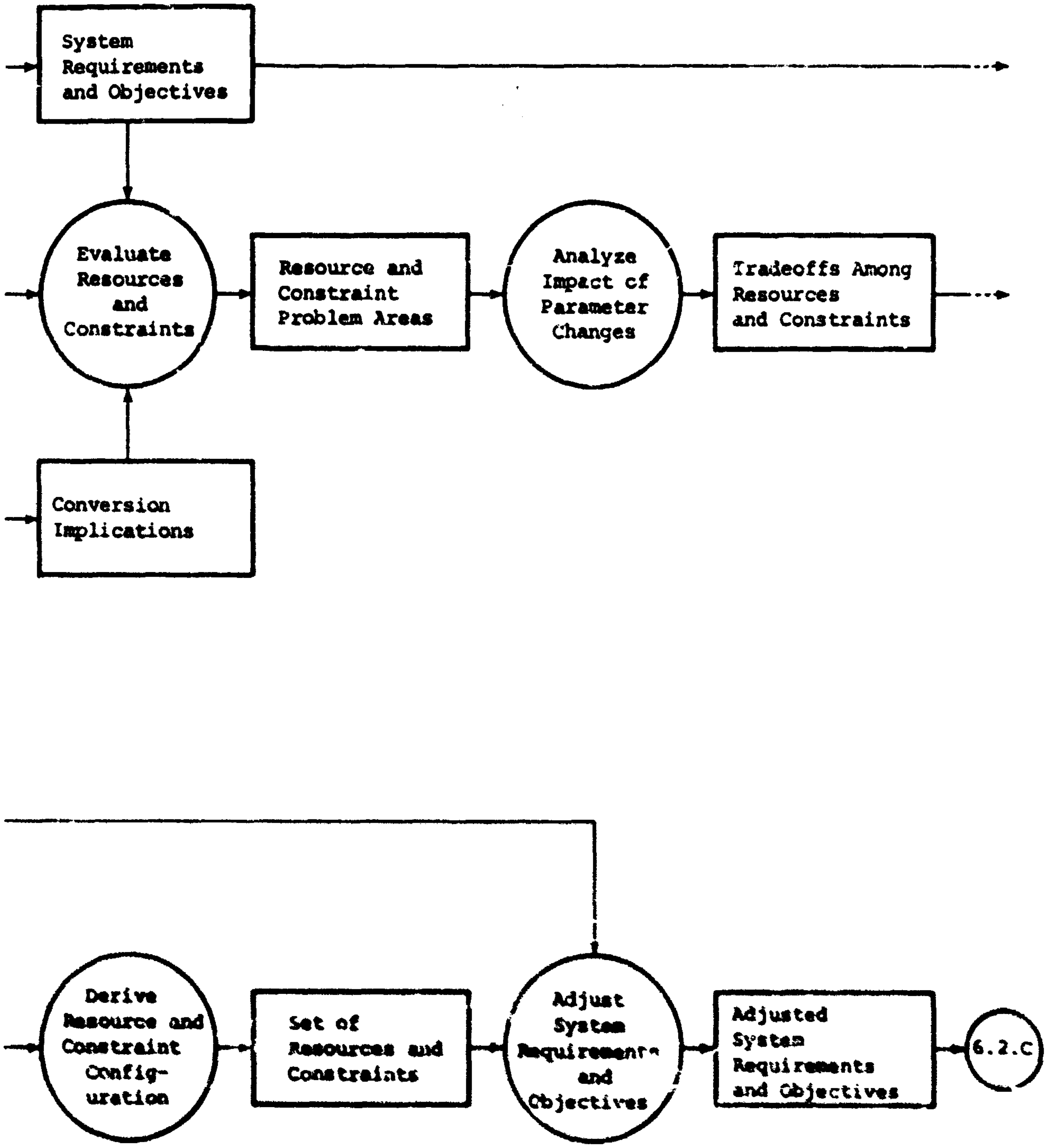


Figure 6-11. (Continued)

Major Relationships Among Resources and Constraints. Each resource and constraint should be examined for its relationship with other resources and constraints. Resources and constraints also have some interactive effects on system performance characteristics. For example, lack of a particular kind of input information can have implications for level of personnel skills and knowledges. Once related resource and constraint types are identified, a network or matrix form may be suitable for depicting the chain of these relationships. This can take the form of a mileage chart on a map, with resources/constraints listed vertically down one column and horizontally in another line. Within this structure, a match point indicates an interrelationship; a no-match point indicates a lack of interrelationship. A mathematical set of symbols can also be used to indicate the degree of interrelaships. Since it is extremely difficult to portray graphically the intricacies of both type and extent of interaction, a narrative form is probably the most appropriate for at least some categories of resources and constraints.

Resources and Constraints Evaluation. Having identified parameters, likely levels they will assume, and closely related clusters, it is possible to estimate the probable impacts of individual parameters and closely related clusters of parameters on the developmental effort. Comparison of stated requirements and objectives with resources and constraints permits identification of sensitivity requirements, critical parameters, imbalances, and potential areas for adjustment. Explication of resource and constraint implications for the conversion from old to new system and for the operational phase is helpful to almost all of the major design stages which follow. The pattern of resources and constraints can be adjusted to accommodate the special requirements of the temporary, but critical, period of conversion.

Tradeoffs Among Resources and Constraints. As a result of identifying critical parameters, the noncritical or less important parameters of resources and constraints are also identified. These noncritical parameters become the candidates for tradeoff analysis. The tradeoff analysis is intended to blend, mold, and reshape the resources and constraints information toward optimization of the new system. Reshaping or reconfiguring these resources and constraints toward optimal alignment with system objectives is the pivotal point of the system development or design process.

Each tradeoff candidate should be translated back into source type of information, i.e., reflected in its original context, so that the change is viewed in proper perspective. This is necessary to assure that the composite analysis has not distorted a parameter to the point where it appears unimportant when it is, in fact, very important.

Experience indicates that if the resources and constraints analysis is not thorough and accurate, either the development effort or the operating system will have built-in surprises. Often in the past, these surprises have been of the negative type--either the development or the operation of the system was not possible or did not live up to its performance expectations.

System Requirements and Objectives Adjustment. If the resources and constraints analysis reveals problem areas which have to be alleviated by resource and constraint tradeoffs, adjustments to system requirements and objectives may be required. The adjustments should adhere as closely as possible to the original requirements and objectives configuration. Once the system requirements and objectives are appropriately adjusted and a set of acceptable resources and constraints is formed, the design activities focus upon describing system functions.

6.3 Identifying Functions (Stage III)

Identifying functions and function relationships is the conceptualization of the minimal processes required to transform inputs into outputs. The functions analysis involved is of system output requirements, input capabilities, and process abstractions. It is not an analysis of hardware nor of existing software routines. The data generated through the identification of functions provide the informational base on which the specific design is accomplished--the allocation of functions to hardware, software, and personnel subsystems. The principal activities in the identification of functions are represented in Figure 6-12.

A function is defined as an action or process. It stands alone or subsumes a number or series of smaller functions or subfunctions. A function

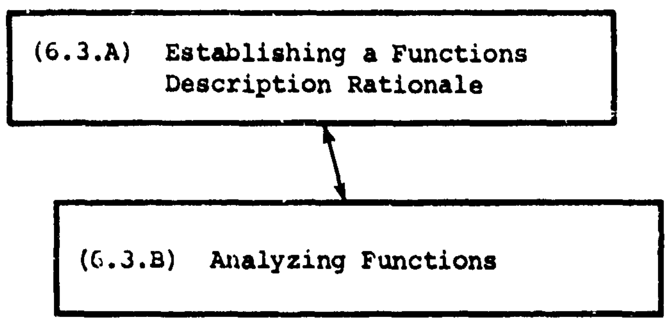


Figure 6-12. (6.3) Identifying Functions (Stage III)

or group of functions can be viewed as an inherent performance characteristic or capability of an entity or thing. It implies a logical rule or set of rules applied to an entity which possesses characteristics fitting the rules.

At its most gross level of conceptualization, a function is the minimum process required to produce a fixed output. At its most detailed and elaborate level, a function is defined as the process which converts the designated input into the designated output. At this level, a complete process statement is achieved. In discussing functions in relation to system descriptions, the terms "function" and "process" are often used interchangeably; "function" is often used to identify the function per se and also the inputs and outputs attached to the function. For purposes here, the words "function" and "process" refer only to the action required. When the thing to be acted upon and the result of that action are attached to the function or process, a process statement is formed. A process statement is then defined as: Input-Process-Output.

As the system development continues and as the functions are analyzed, process statements refine and specify input, process, and output associations in the system. When the system is fully developed, a detailed description of every process applied in converting every associated input into the required output is possible, i.e., a fully detailed functions description of that specific system configuration. All of the process statements resulting from functions identification and analysis together form the functions description of the system.

6.3.A Establishing a Functions Description Rationale

Establishing a rationale for functions description includes identifying functions information sources, determining means for presenting that information, and developing strategies for conducting effective and efficient functions analysis, as shown in Figure 6-13.

Sources of Functions Information. These sources are identified and discussed briefly below.

The system information to date. This consists of system requirements and objectives coupled with resources and constraints. In

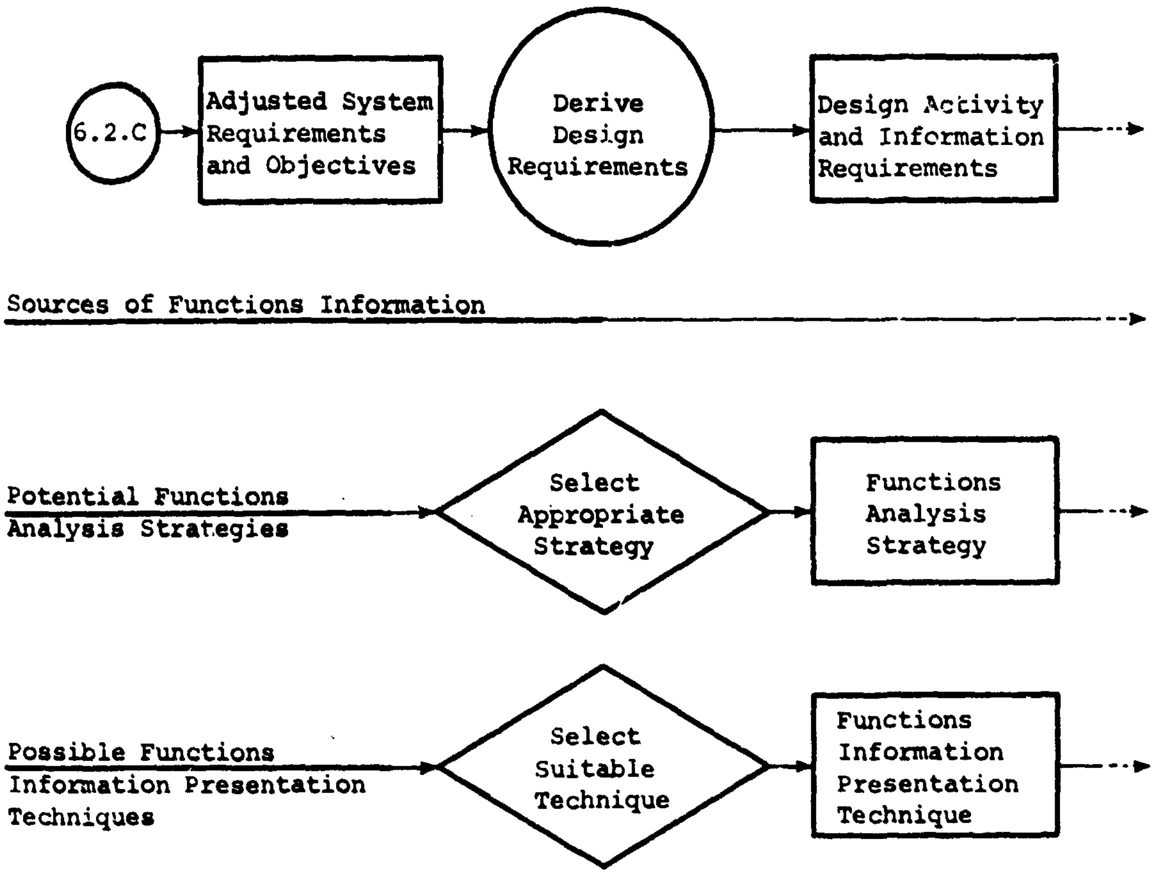


Figure 6-13. (6.3.A) Establishing a Functions Description Rationale

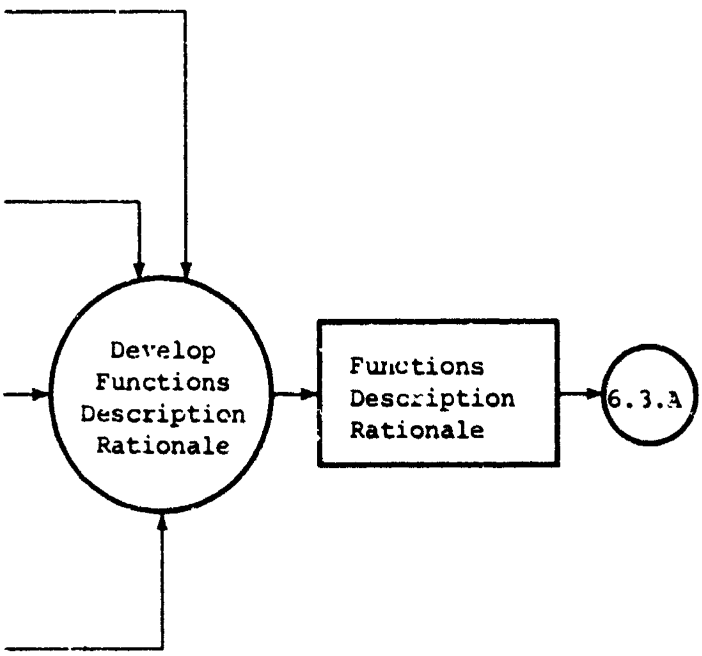


Figure 6-13. (Continued)

general, output requirements are identified through the requirements and objectives while the resources and constraints provide information concerning inputs. To the extent that this information relates inputs to outputs, process abstractions are achieved.

System lineage information. Many current system design efforts involve reallocation of processing. However, use of current system description should be approached cautiously so that imitation does not replace design. Characteristics of the system inputs and outputs associated with earlier versions of the system should not be assumed or saddled to input and output characteristics of the developing system. Only when characteristics are proven essential or appropriate to the new system should they be attached.

Subsystem "chunks." Where earlier developed systems are assumed in the developing system as subsystems, the functions description of these "chunks" is incorporated into the system functions information. The particular relevance of this information is in achieving an appropriate interface between the function subsystems.

Other similar systems. Those functions currently producing outputs which correspond to the developing system output requirements also contribute to the functions description information, particularly if the given inputs remain the same. However, the analysis of an existing system can exert a tyrannical influence on the design of the new system. For this reason, constant attention must be given to function alternatives rather than to slavish recapitulation of functions found in related systems. On the other hand, failure to draw upon experience gained in the previous development of related systems results in needless waste.

External information. Descriptions of the kinds of processing within man and machine capabilities and the logical relationships between these are derived, in part, through search of human factors and machine processing literature. Such information is most appropriate where new capabilities are structured. Manufacturers' routines for processing specific categories and types of information

should be included in the data base. Functions descriptions of machine capabilities--their structure and sequence--define the limits for functions descriptions appropriate to use context.

Limits on man's processing capabilities are continually defined in terms of new requirements imposed by other than routine manipulation of data. Although direct insertion of this type of data from one use context to another cannot be assumed, much is gained through its interpretation and through structured predictions of how altered conditions will affect processing requirements.

Design Activities and Information Requirements. Major differences are readily observed in information requirements imposed by the varying levels of design on the functions description of the system. A more substantial amount of information is required if training support materials are a system-design consideration. The need for additional information also implies that different types and presentations of information are necessary for communication with training personnel.

The interface of the developing system with other systems and system components also bears upon functions identification and analysis. In a system which requires no interface with other fully developed systems (or partial systems), establishing and meeting information requirements is a straightforward task. However, more usual system development efforts must respond to interface requirements--fitting the developing system into other systems to provide a cohesive operation, or fitting other smaller systems into the developing system as function subsystems. Unless system design is aligned with presently available "chunks" incorporated into, or interacting with, the system, the requisite interface is difficult to achieve.

Functions Analysis and Information Presentation Strategies. The major user of functions information is the design team. Their requirements are paramount, and the extent and type of information presentation must permit and facilitate design activities. Other important uses of the functions information relate to management requirements. The decisional role of management personnel must be supported by adequate functions information so that

decisional alternatives can be evaluated in terms of objectives and requirements and the known resources and constraints. The functions information often supports description of function subsystems--the grouping of like functions into composite units--for evaluation and selection of major system components. A complex system which is designed to satisfy many major operational objectives may require a mission analysis or contingency analysis. Such a requirement imposes additional information handling and presentation requirements on the functional description.

It is impossible to disassociate techniques for presenting functions information from the strategies used in functions analysis. The technique should be selected on the basis of how well it facilitates achieving the stipulated output requirements of the analysis strategy. Requirements which are directly associated with selection of the presentation technique include:

1. The capability of the technique to support all requirements related to design and documentation.

2. The capability of the personnel to use the presentation technique, both in recording and communicating information.

3. The organizational capability to produce and transmit the documentation in sufficient quantity and in accordance with schedule requirements.

Whatever its selected format, the functions information provides a network of system descriptive information. The preferred means and specific design of formats depend on the specific system and the developmental requirements. The principal means for presenting functions information are:

1. Narrative text.

2. Flow diagrams.

3. Event networks, e.g., PERT.

4. Mission descriptions.

5. Simulation models

6. Time-line charts.

7. Adjunct lists and tables.

8. Decision and contingency tables.

9. Combinations of the above.

Standards for the Rationale. The steps of functions analysis are a broad conceptualization of the strategy. There are limits to which these step activities are dictated. It is desirable to incorporate considerable latitude in the functions analysis strategy, and to select the strategy on the basis of output rather than the procedures applied. However, some minimal set of standards needs to be established so that communication does not break down. The output of the selected strategy should permit:

1. A basis for identifying functions information at a requisite level of detail.

2. A basis for communicating design information.

3. Ease of converting data base information into functions descriptions based on:

   a. The format and status of other system information and relevant interface or environmental information.

   b. Eliminating redudant analysis, i.e., identifying common function sequences.

   c. The maximum utility of varying levels of functions information in establishing input-process-output requirements.

   d. Sampling among contingencies and predicting from that base.

6.3.B Analyzing Functions

Functions analysis is an evolutionary activity which results in a functions description of the total system to a level of detail and accuracy which permits:

1. Allocation of the functions, including evaluation of alternative allocations where appropriate.

2. System design to the designated level of detail.

3. Description of the system in accordance with selected categories of information, such as:

   a. System models.

   b. Function subsystems.

   c. Component subsystems, following allocation of functions.

   d. Mission contexts.

4. Derivation of requisite formal system documentation.

A representation of the activities involved in functions analysis is presented in Figure 6-14.

Process Statements. In functions analysis, system inputs are matched against outputs to identify the requisite functions or processes which transform the inputs into outputs (process statements). These process statements are the "what" of the system operation which are translated by their allocation into the "how" of the operation. The functions analysis is an attempt to determine what needs to be done (the process) to what is available (the input) to produce what is needed (the output).

Alternative strategies available for establishing the nature of process statements required include:

1. Generate process statements to the level of detail which fits the definition of early systems design.

2. Generate process statements to the level of detail possible without entering into "contingency" situations.

3. Generate process statements to the level of detail which allows the description of alternative man-machine capabilities.

4. Generate process statements to the level of detail where the evaluation and selection of alternative capabilities, i.e., the allocation of functions, can be made.

These are not exclusive categories, and some combinations of these or other statements may be more appropriate to the specific system-design effort.

Input characteristics. It is important early in deriving process statements to establish a methodology for defining and communicating, i.e., working within, a set of input characteristics and parameters. Obviously, there are many parameters of information which are used to describe input requirements in detail. However, a set of information description characteristics and parameters has not been developed which universally applies to the description of information system inputs. The appropriateness of the set of characteristics is, to some extent, inherent in the type of information handled. Possible information system input characteristics include the types of information, physical characteristics of the information, number and format of code elements, format of the information, volume of information, frequency of inserts, span of information, insert time, presentation medium, and quality of information.

Output characteristics. Since the input, acted upon by a process or function, produces the output, the same set of characteristics and parameters used to describe the input is applied in describing output characteristics.

Processing requirements. The processing requirements of the system are derived from the processes or functions that convert every associated input into the required output. Identification of system processes and functions depends, in turn, upon detailed examination and analysis of the input and output requirements of the system. As process statements are formed, Input-Process-Output, a detailed description of the system processing requirements is produced.

Mission Analysis. For large and complex systems, where a diversity of system objectives is identified, a mission analysis structure provides a useful analytic tool for system development activities. The purpose of

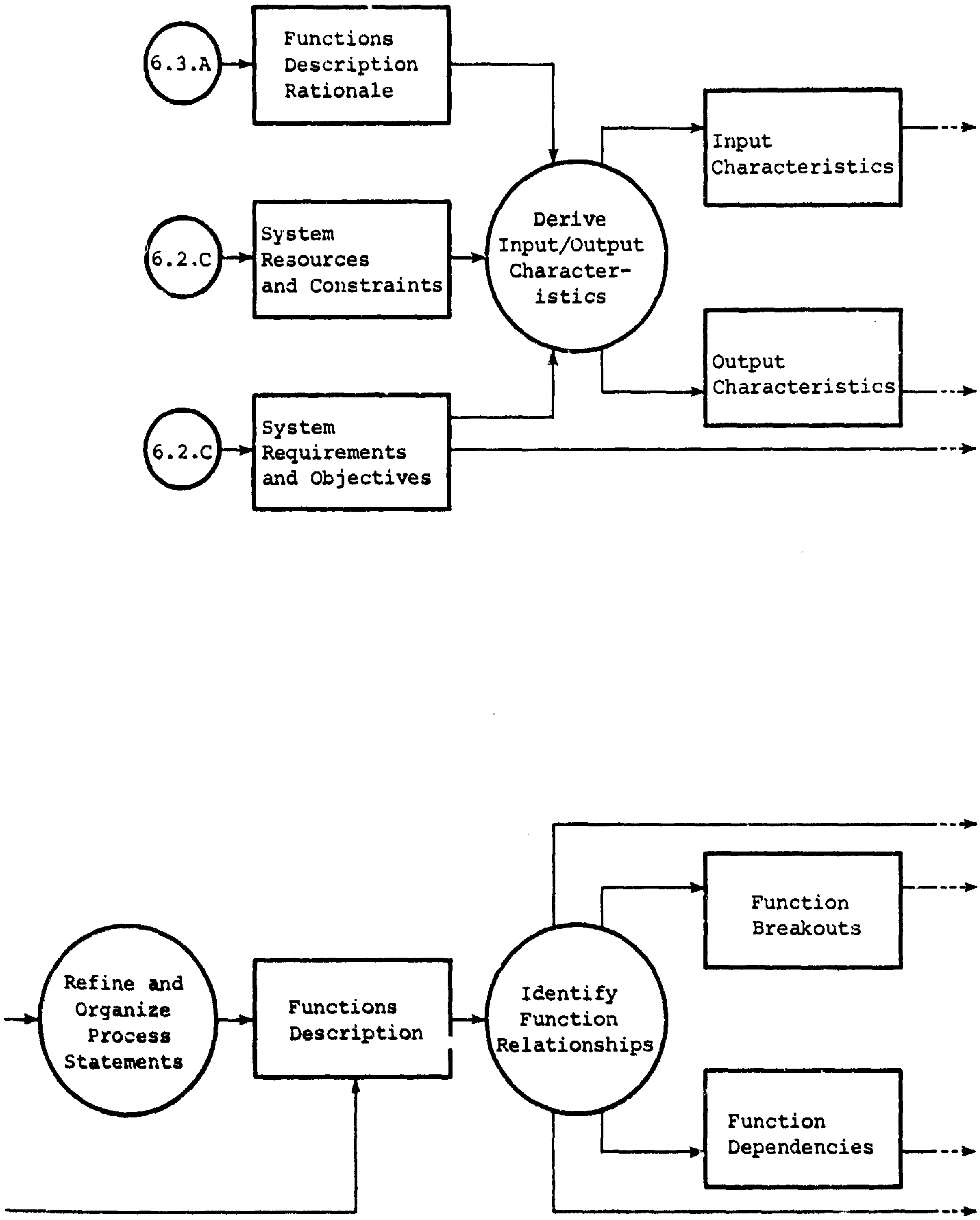


Figure 6-14. (6.3.B) Analyzing Functions

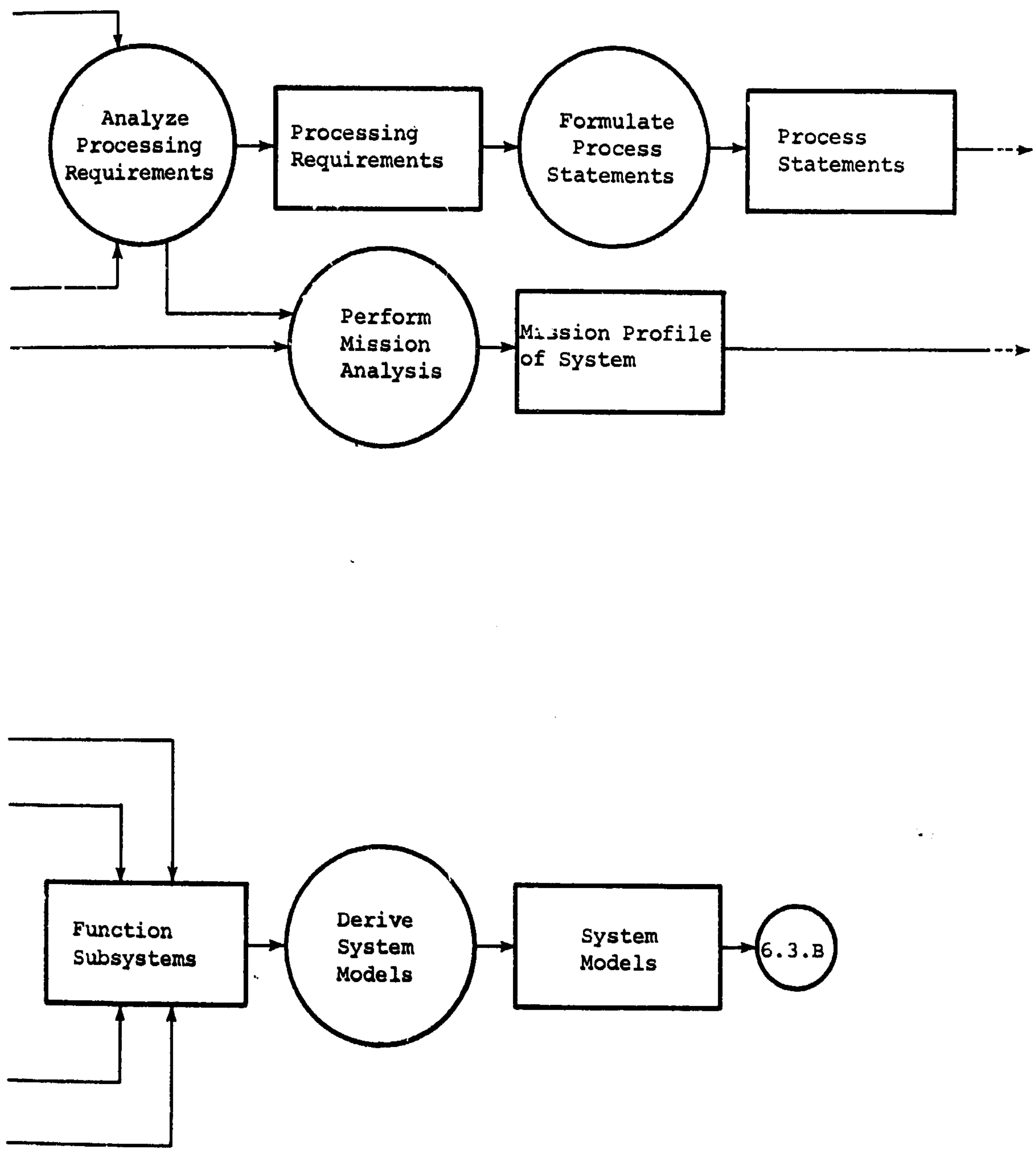


Figure 6-14. (Continued)

mission analysis is to assure that the system is designed to attend to all operation contingencies. It specifies routine contexts of operation that the system must be designed to handle, as well as those of lesser frequency or importance which must also be accounted for in design. Massive contingencies tend to cause sufficient impact on performance requirements for the system so that it is necessary to define separate types of missions. Any characteristic mode of system operation can imply a separate type of mission.

Mission analysis involves a conceptual walkthrough of a representative sample of the universe of missions appropriate to the system. By depicting and grouping mission segments, the analysis demonstrates the effects of meeting one mission type on the ability of the operational system to meet other operational objectives.

On the basis of the data base, in particular the requirements and objectives, it is possible at this point to select and subdivide one or more representative mission cycles. The derived mission segments are then sequenced to provide meaningful mission profiles with which functions breakouts can be associated. In the final steps of functions description, mission scenarios or simulations become relatively sophisticated--involving descriptive, analytic, probabilistic, or mathematical choices of conditions, or determinate choice of condition models. These simulations provide a convenient tool for the evaluation of the proposed system design to meet operational requirements under all contingencies of operation.

Functions Description. At any level of detail in preparing process statements, a functions description of the system is possible. The functions description of the system includes function breakouts and function dependencies which are derived from process statements and incorporated in function subsystems and system models. Function breakouts and dependencies are described below:

Function breakouts. A function breakout is a description of system functions which reflects the hierarchical nature of the functions. Function breakouts are derived through the iterative activity of refining stated functions. Throughout the system design process, it is possible to generate increasingly refined function breakouts,

each breakout permitting identification of a total function at that level. To the extent that required output characteristics are known, it is usually most effective to begin at the terminal output and work backward. However, in some cases, deriving function breakouts is facilitated by working from both output and input ends toward a hookup within the system. In other instances, it is necessary to derive output characteristics by analyzing requisite hierarchies of functions which operate on input characteristics.

Function breakouts or hierarchies are considered in determining the content of process statements. For example, if the preparation of training materials is a part of the design effort, it is necessary to describe in detail all hierarchical effects of the statements so that cohesive training packages can be designed. Examples of guidelines for the amount of content relevant to this perspective include:

1. Generate process statements which depict functional relationships (hierarchies and associations) to their logical beginning.

2. Generate only derivative process statements, i.e., statements which define system functions, not which build an upward hierarchical chain.

3. Generate process statements which identify all input/output characteristics whether that characteristic can be specified or not.

As with the process statement strategies previously noted, these are not exclusive categories. Some grouping of these or additional categories may better fit the specific design requirements.

Function dependencies. Function dependencies demonstrate some associative interaction of the functions, either through identification of a common element or through the derivative nature of function hierarchies. Types of function dependencies of utility to the functions analysis include:

1. Dependence upon some common input, output, or process requirement, or characteristic(s) of these.

2. Dependence upon some common time base.

3. Dependence upon some common spatial consideration or relationship.

4. Dependence upon some logical relationship.

5. Dependence upon some logical derivative, e.g., the hierarchy of functions.

Function Subsystems. Grouping process statements into constellations of related functions permits the identification of function subsystems. These groupings are of great utility in the efficient allocation of functions to subsystems (hardware, software, and personnel). When the allocation of functions is made, regrouping functions on the basis of their associations with hardware, software, and personnel functions is possible. For large systems, and where design requirements encompass all aspects of system design, these detailed function subsystem descriptions are advantageous in the design of equipment modules, software routines, training packages, etc.

System Models. Relatively early in the process of functions description, it is desirable to formulate a tentative general model of the system--subject to modification and detailing as the breakout of functions continues. Although preliminary system models, either function or mission oriented, are usually somewhat disarticulated, they converge into integrated frameworks as the functions description continues. The type(s) of system modeling chosen should be relevant to the documentation requirements of the system-development program. These requirements are focused on in the selection of the functions analysis strategy and presentation technique.

In addition to directly supporting the systems design effort, documentation of the systems information provides information in a format appropriate for and relevant to:

1. Upper echelons of system development personnel monitoring and directing the system design process.

2. Management personnel who must concur with and lend approval to the system design effort.

3. System design personnel who must perform the system-design activities.

For all of these purposes, more than a simple historical presentation of what has occurred to date should be presented. Firm design decisions and their implications are shown. Decision points and their alternate choices which can be identified are clearly presented within their responsibility context.

6.4 Allocating Functions (Stage IV)

Functions allocation is the process of assigning functions to the hardware, software, and personnel subsystem best qualified to handle that responsibility. Through the allocation of functions, a specific system configuration is identified, i.e., the allocation of functions is the terminal design point of the early systems design effort. In that sense, the allocation of functions continues through the design process until the "last nail is driven."

Identifying requisite functions for the system cannot proceed without considering "how" the functions are to be achieved. Without such intermediate allocation considerations, analyses of all available functions allocation contingencies would be required--hardly an efficient systems design method. The principal activities included in allocating identified functions are shown in Figure 6-15.

For allocation purposes, man, machine, and software subsystems are defined and characterized as follows:

Man (Personnel)--the human component(s) of the system, including skills and knowledges, experience, training, and selection. The major roles of man in a man-machine information system are manipulation, control and coordination, and decision making. Man's greatest asset is adaptability to new or changing information contexts. Men are dynamic because they never repeat a given state; they are

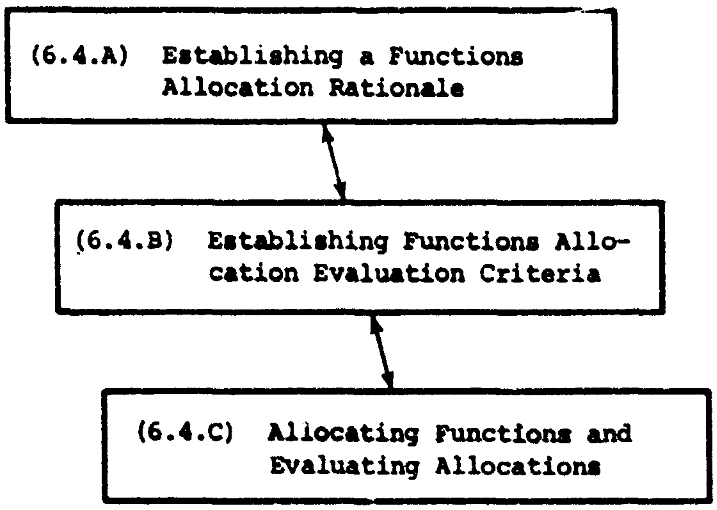


Figure 6-15. (6.4) Allocating Functions (Stage IV)

autonomous because they can act independently of other system components (although such a condition is rarely found in a man-machine system). Their energy is self-contained--man can function as either a total system or as a system subsystem.

Machine (Hardware)--the machine component(s) of the system. Included are all those "hard" pieces which, through design, become an integral part of the man-machine information system. The major role of hardware is to carry out controlled instructions directed either by man, software procedures (programs), or these two in combination. The machine's greatest asset is that it can rigidly repeat a given state. It may contain built-in, simple routines and can, through instructed use of these routines, simulate a dynamic state. It is dependent, but again through instruction, can simulate an autonomous state. Machines have no self-contained energy but use energy to convert information (or matter) from one form to another. Machines cannot operate as self-sufficient systems, although they may simulate a total system.

Software (Programs)--those "pieces" which are auxiliary or supportive to the personnel of the system--although their application, closely associated with hardware, can be viewed as subordinate to machines rather than personnel. The software portions of the system are storage units for personnel and of decreasing utility as man's capacity to store information increases. (For some types of information, e.g., computer control manipulation, such man-storage rarely occurs.) Software is totally static and dependent in that it cannot be changed and retain identity and cannot initiate change in itself. No energy is either contained or used--the role of software is to direct the energy of the personnel and hardware subsystems through information content. Software cannot be or simulate a self-contained system; and even as a system subsystem, it is subsidiary to the personnel and hardware subsystems.

6.4.A Establishing a Functions Allocation Rationale

It is desirable to expand the development rationale commitments formulated for other design stages to the functions allocation stage. The rationale covers at least the following issues:

1. The capabilities and limitations inherent in the system design requirements.
2. The capabilities and limitations of the design personnel and other developmental resources.
3. Rules for generalizing information from external sources.
4. The identification and selection of an appropriate strategy for performing the allocations.
5. Criteria by which optimum allocation can be evaluated and tradeoff models which guide the allocation.

The areas of information which contribute to establishing the functions allocation rationale are shown in Figure 6-16.

System-Relevant Considerations. System design implications relevant to the functions allocation rationale include:

1. System design requirements which dictate the allocation of specific activities to man, machine, or software.
2. System design interface requirements.
3. Inclusion of previously development system "chunks" as components of the system.

Implications from the interface and system "chunks" information are broader than their own specific allocation limitations. That is, they contain implications for consistent allocation of similar functions and for functions allocation dependencies.

Development Requirements. Identified resources and constraints are the primary source of information for development requirements. Particularly relevant categories of information are:

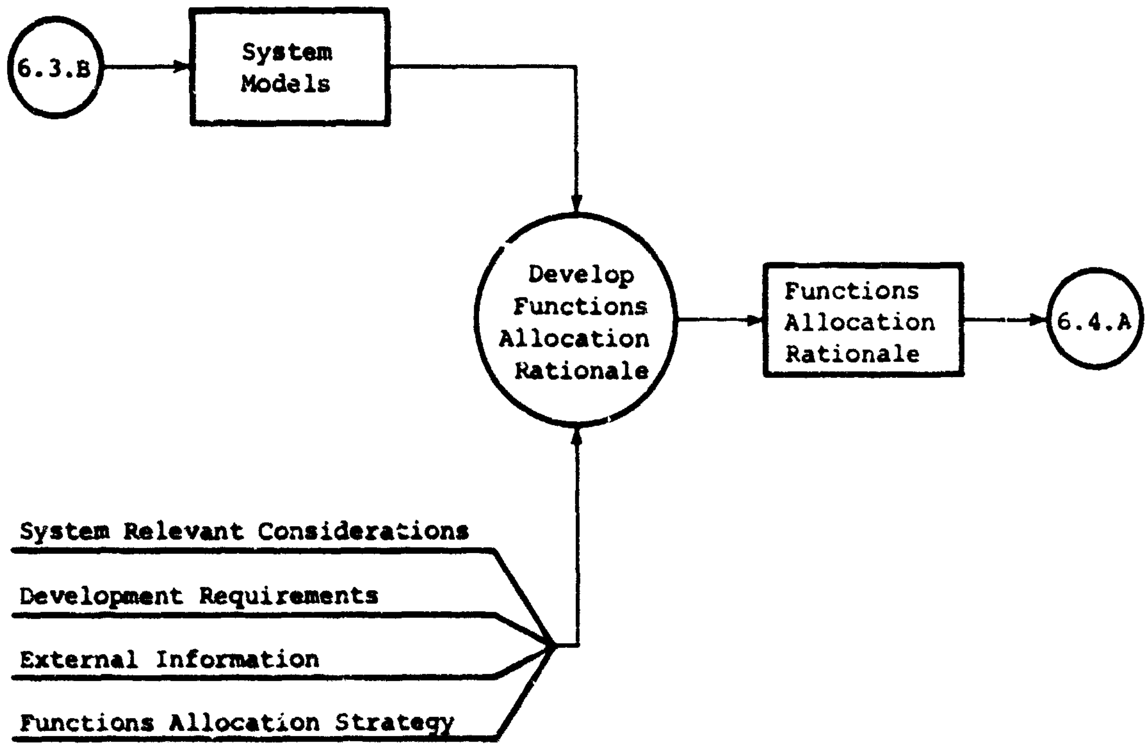


Figure 6-16. (6...A) Establishing a Functions Allocation Rationale

1. Design group capability--the types and numbers of personnel available to participate in the system design activities.

2. Scheduling requirements.

3. Cost considerations.

The utility of the resources and constraints information is for establishing the limits of development activity. In effect, these implications identify GO/NO GO design situations in regard to the amount of machine design or programming development, system design personnel training, etc., required under alternative functions allocations. A rationale which establishes development requirements is more generalizable to the total system development picture than to specific aspects of that development. It creates a balance sheet or tradeoff model for the resources and constraints in relation to alternative allocations.

External Information. Areas of information external to the system-development information which have a bearing on the functions allocation rationale include:

1. Generalizable man-machine capabilities.

2. Other systems design.

3. Hardware and software informations.

The human factors literature is replete with versions of the "Fitts List," i.e., identification of types of performance which are best assigned to man or machine. Such lists have more significance in establishing a rationale for the allocation than in actually performing the allocation since they contain generalized comparisons of these two capabilities. Other human factors literature which examines the "limits" of man's capabilities, e.g., laboratory study results, can be used in defining a rationale if conditions of use, design considerations, and experimental design information are detailed enough to be applied to the developing system. Overgeneralization is a hazard here, particularly when this information is used for establishing a rationale. The conflicting statements available in the body of human engineering design

information indicate that design and use context details play a considerable role in assessing the utility of the information to specific design situations.

Reference to functions allocation rationale statements from other similar design efforts also provides useful information. Again, it is important not to overgeneralize this information and thereby impose greater restrictions than necessary on the freedom of functions allocation. The general body of data processing and retrieval literature, the equipment manufacturers' literature, and their software routines produced for use with that equipment contribute effectively to the definition of an allocation rationale. These informations verge on the actual allocation of functions and should be interpreted only as contributing to the functions allocation rationale.

Functions Allocation Strategy. There are no easily identified sources that describe strategies applied in the allocation of functions. This activity--a part of the design process only indirectly reflected in the final system design--does not usually become documented in design information. The strategy used is of necessity system-specific, so the amount of information which can readily be transferred from one design situation to another is often minimal. There are, however, procedural considerations which can be made. These relate to:

1. The level of allocations.

2. Priorities in making the allocations.

3. The paths of allocation, relevant to the hierarchical nature of the functions.

4. Decisional routines which should be applied.

5. The extent to which allocation should be interspersed in deriving process statements.

6. Determining an appropriate interface between allocation and evaluation, in terms of:

   a. When the evaluation occurs.

   b. How the evaluation is made.

c. Routines for modifying process statements or entering new process statements which result from allocations.

6.4.B Establishing Functions Allocation Evaluation Criteria

A set of criteria for evaluating individual and collective allocations must be developed prior to actually making the functions allocations. These criteria are, of course, subject to revision and modification as alterations are made in system requirements. They are of greatest utility in assessing alternative allocations (or capabilities) and in forming a baseline set of data for generating system test and evaluation criteria used in further system design decisions. To the extent possible, they should be applicable to both evaluation of alternative capabilities at a process statement level and at the collective functions allocation level. That is, the criteria should not be generalized to only the final system performance requirements, since no evaluation would be possible until the complete system is configured. Detailed evaluation criteria are applied in selecting optimum allocations when allocation decisions must be made. The information employed in establishing criteria for evaluating functions allocations is shown in Figure 6-17 and described in the following sections.

Early Design Termination Criteria. The allocation of functions to man, machine, or software is the final active design effort of early design. Later stages, describing the design concept and determining design feasibility, do not further refine the system design. Then, to establish criteria for evaluating functions allocations, it is appropriate to distinguish characteristics of early design which set it apart from later stages of design. Some of these differences are identified as follows:

1. Tooling equipment, preparing machine programs, describing personnel tasks, preparing training requirements, etc., are not included in early system design.

2. In early system design, requisite (man-machine-software) capabilities for effecting the system design are selected rather than actually employed.

3. In early system design, adjustments can be made in system requirements and objectives.

Because process statements are prepared at all levels of design, the criteria used in defining early design termination are open to individual interpretation. It is a systems design management responsibility to set a fine definition of the criteria and interpret and communicate it to personnel who must work within its limits. However, a basic set of statements concerning the final design level of early design is:

1. System functions have been identified and described.
2. The development requirements in meeting the system descriptions (functional and component) have been identified.
3. The identified system functions have been allocated to hardware, personnel, and software.
4. The effect of alternative function allocations on capabilities to meet system development requirements has been considered in evaluating and selecting the available allocation alternatives.
5. When necessary, alternative allocations of functions to subsystems have been evaluated and the best set of alternatives has been selected.
6. The allocation of functions has permitted the identification of personnel, hardware, and software subsystems.

System Performance Evaluation Criteria. Criteria for evaluating system performance include:

1. Schedule considerations.
2. Maintenance requirements of the system.
3. Acceptability of the system:
   a. To the system operators.
   b. To the system users.

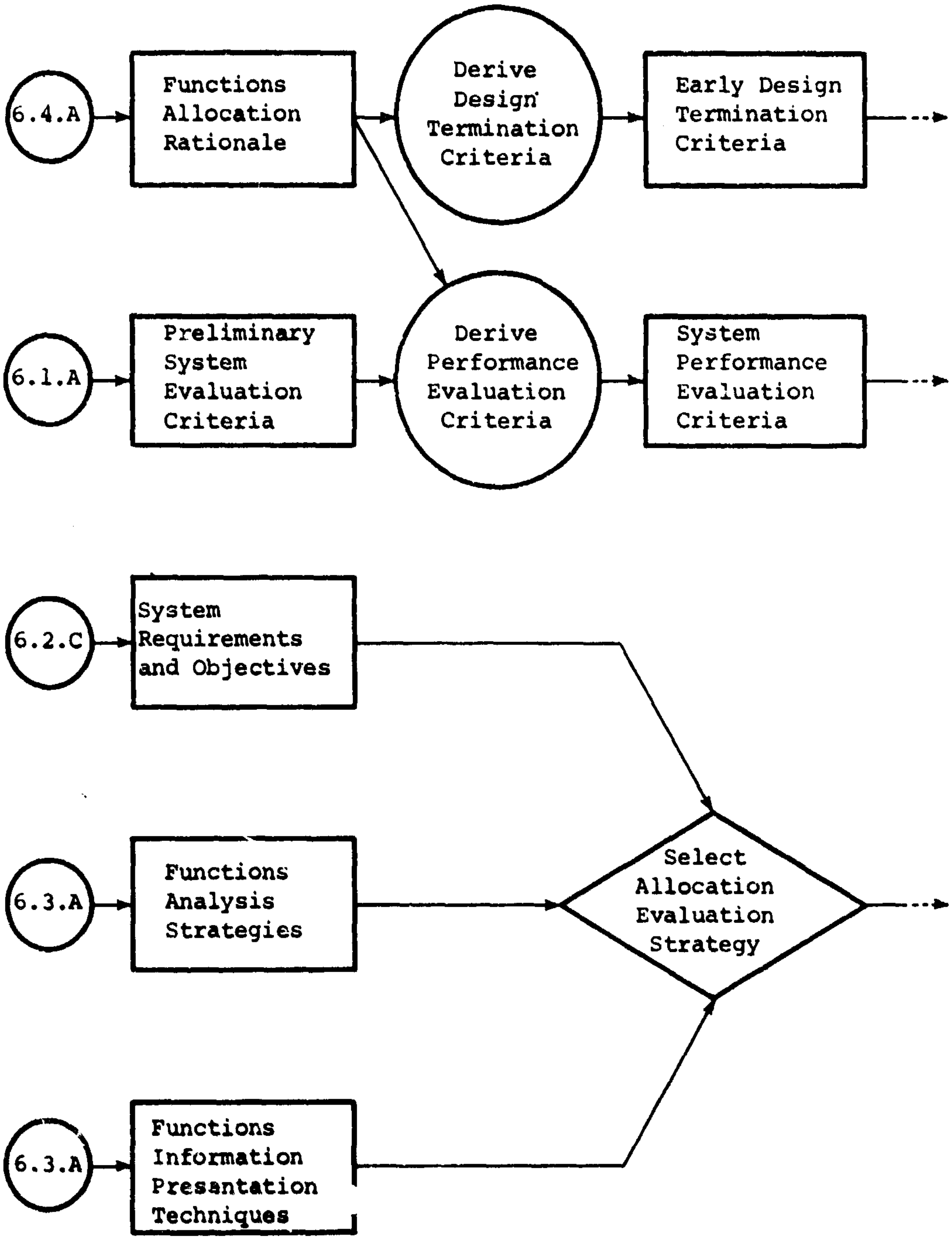


Figure 6-17. (6.4.B) Establishing Functions Allocation Evaluation Criteria

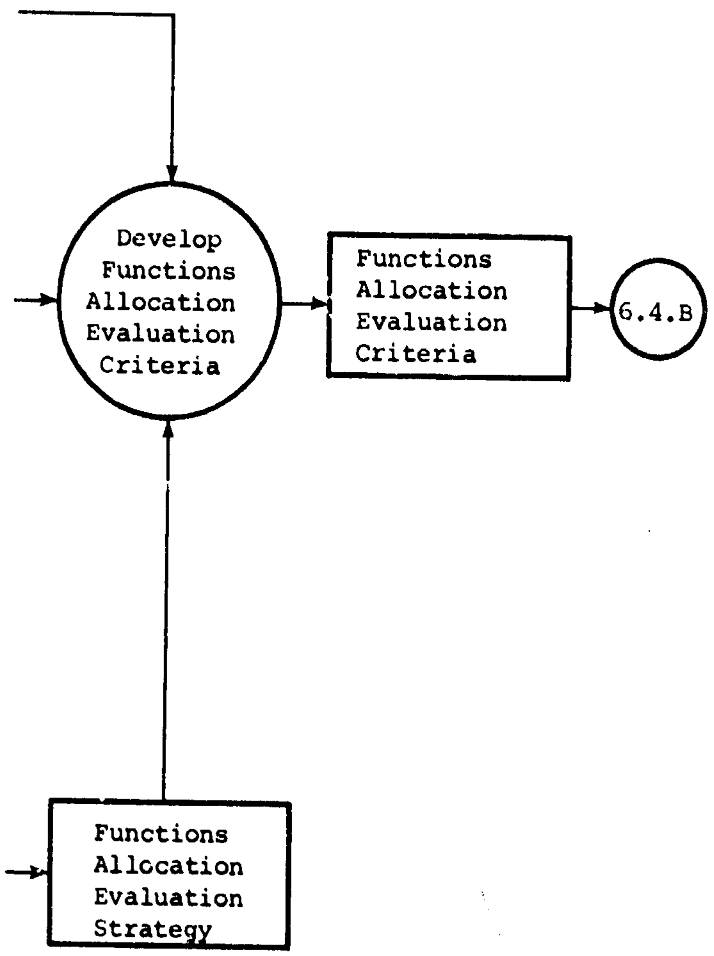


Figure 6-17. (Continued)

c. To higher management:

(1) Systems design management.

(2) Company management.

4. Regulatory considerations, e.g., FCC.

5. Conversion and add-on considerations.

6. Effect on other company operations.

7. Educability:

a. A statement of requirements to update user and design personnel capabilities.

b. A statement of the limits of user education and design personnel in new techniques, etc.

8. State-of-the-art, a statement of intents to and limits in furthering state-of-the-art.

Criteria suggested elsewhere in this handbook (Design Assessment and Systems, Process, and Products) and from the specific design context are also appropriate.

Once system performance evaluative criteria are established, it is necessary to assign priority levels to them in order to differentiate "critical" and "not so critical" system and development requirements. These priorities become the mediators for selecting alternative allocations of functions once the evaluation stage is entered, i.e., they are used to weigh tradeoff effects between the allocation alternatives.

Functions Allocation Evaluation Strategy. The type of evaluation strategy appropriate for evaluating the allocation of functions is very closely tied to the selected functions analysis strategy and functions information presentation technique. The strategy must permit the projection of design information to performance expectations, and it must allow for the evaluation of alternative functions allocations. The system models--functional and mission oriented--provide a good framework for deriving appropriate simulation and tradeoff models.

6.4.C Allocating Functions and Evaluating Allocations

The allocation of functions is the decisional process through which available system component capabilities are exercised against process statements. The evaluation criteria are the mediators of these matchings. Having assembled a rather large amount of system-descriptive information, e.g., system requirements and objectives, resources and constraints, performance expectations, input/output requirements, evaluation criteria, etc., it is obvious that the allocation of many functions is dictated by compelling considerations (hence, dedicated functions). For such situations, it is wasteful to elaborate alternative capabilities for their achievement. For those functions which are not so obviously dictated, alternative capabilities should be identified for accomplishing each function. Caution must be exercised in allocating dedicated functions, however. It is important to avoid overinterpreting dedication and allocating functions on the basis of previous allocation decisions. On the other hand, to ignore dedication implications, or even previous or other system allocations, can result in great and unnecessary expenditures of time in identifying alternative capabilities when the best allocation has already been made. A flow chart depicting functions allocation activities is presented in Figure 6-18.

In spite of well-defined evaluation criteria, the allocation of functions remains a judgmental activity in many instances. Accurate assessment of the conditions for allocation is essential. One characterization of the decision contingencies is presented below:

Dedicated--those situations where, through system requirement or lack of alternatives, no real decisional process can occur for allocating functions.

Optioned--those situations which can be identified, because of other decisions or information, as permitting no "real" decisional process in the function allocation.

Evaluated--those situations where alternatives can be identified and evaluated for selection of the "optimum" allocation.

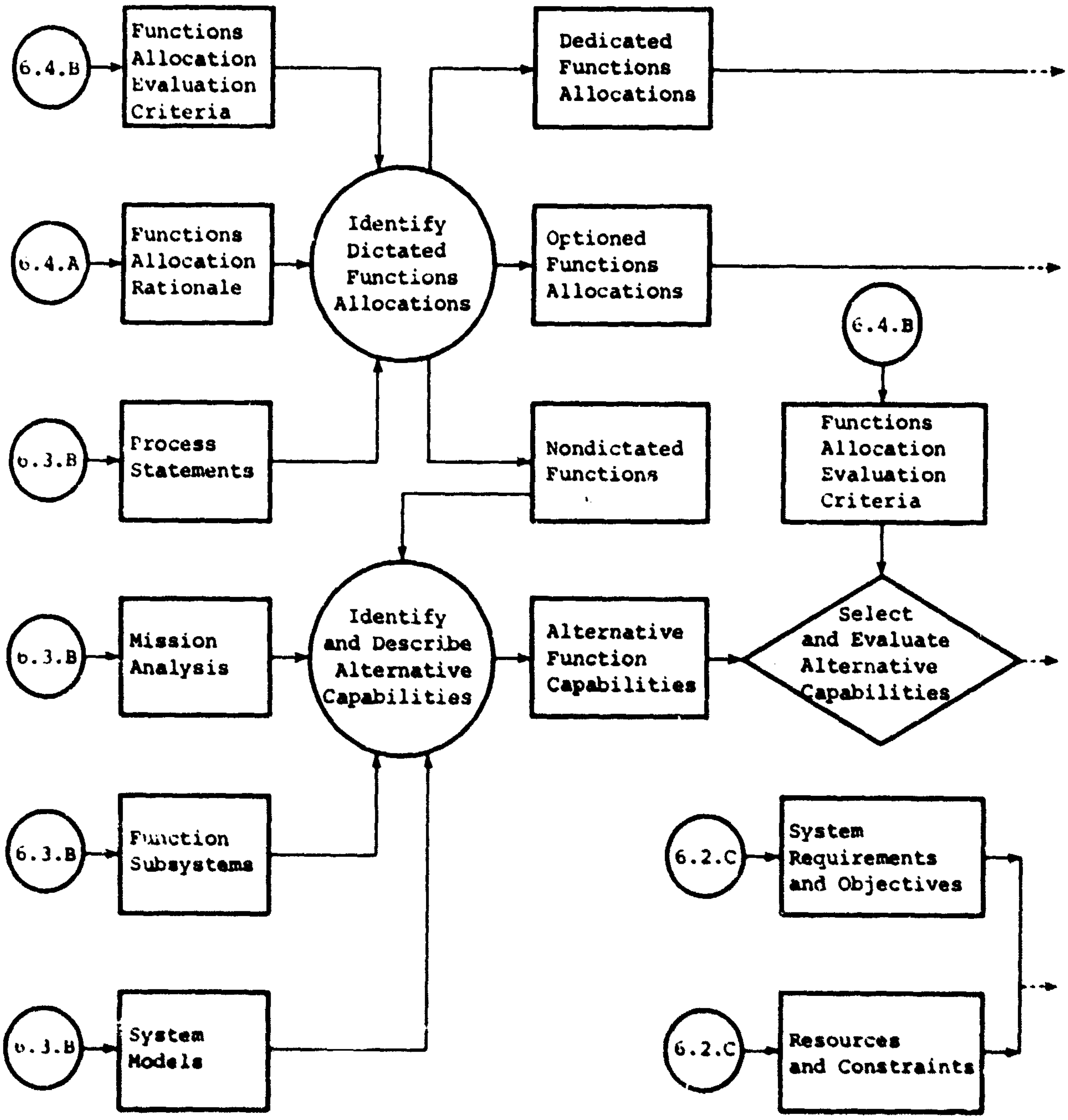


Figure 6-18. (6.4.C) Allocating Functions and Evaluating Allocations

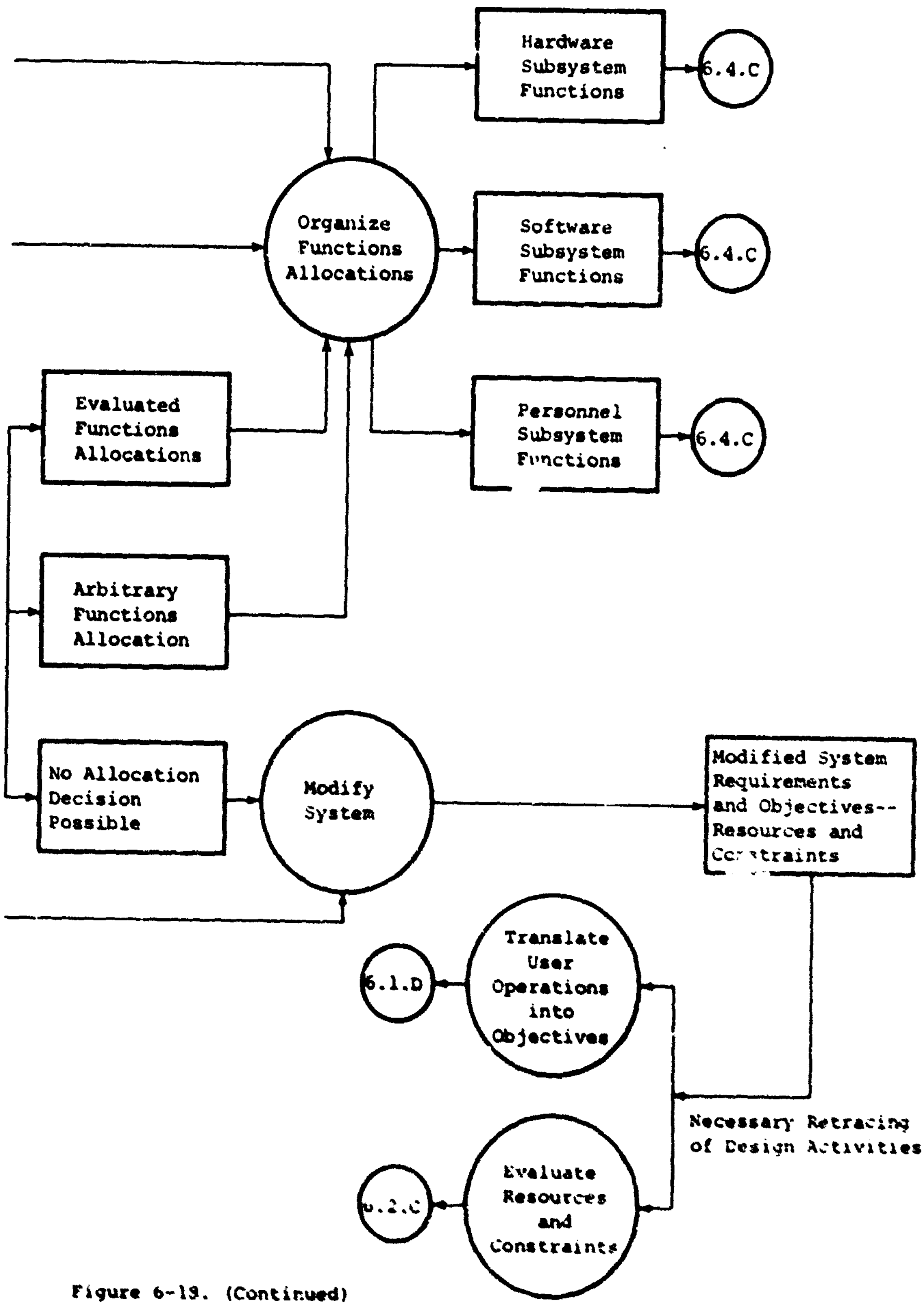


Figure 6-19. (Continued)

Arbitrary--those situations where alternatives can be identified, but evaluation indicates no "optimum" allocation, resulting in a "discretionary" allocation.

No Decision--those situations where no allocation can be made, resulting in required modification of requirements and objectives and/or resources and constraints.

Alternative Function Capabilities. In this context, a capability is any mechanism, entity, or process having the potential to accomplish a system function. For each system function, any reasonable alternatives among hardware, software, and human performance capabilities are identified. Identifying alternative capabilities is a highly intellectualized and system-specific activity. Fortunately, there are limitations imposed on the activity by the existence of previously dedicated functions. It is appropriate to consult system models, function subsystem characterizations, and the mission-analysis information to further limit the identifiable range of possible capabilities. All systems information and those areas of information outside or external to the actual system (man-machine capabilities, machine routines, etc.) also help to limit capability alternatives in functions allocations.

Alternative Capabilities Evaluation and Selection. Where appropriate allocations are not obvious, a formal comparison among the available capabilities is necessary. This requires making performance estimates, generally quantitative, for activities are known at this stage only in terms of their general characteristics. Machine capabilities are well documented. Human performance technology is not sufficiently advanced to support very precise estimates under these conditions. However, by marshaling the most relevant available data, particularly from field studies of similar systems, decisions among alternative allocation possibilities can be made.

Once tentative allocations are made, relatively sophisticated mission simulations establish a basis for estimating the total effects of the allocations. That is, the evaluation criteria are derived which permit evaluation of generalized final system performance requirements and evaluation at the process statement level. Different functions and sections of the system

involve different levels of difficulty and require varying numbers of iterations before satisfactory allocation is achieved. Allocations should be defensible in terms of achievable performance and cost.

System Modifications--Requirements and Objectives and Resources and Constraints. Where no decision points are reached in functions allocations, it is necessary to modify system requirements and objectives and/or capabilities to permit system design within an adjusted framework. Once adjustments to requirements and objectives are specified and their probable effects determined, a decisional activity is required on the part of the systems design management. The resultant modifications should be kept to a minimum and be consistent with overall system objectives. If the modifications exceed these limits, i.e., impinge upon critical objectives or exceed resources, the whole or major part of the design process must be repeated to insure that:

1. The revised objectives or resources are comprehensively accounted for in the design process.
2. The effect(s) of the revised objectives or resources on other aspects of the system development and projected design are accurately evaluated.

6.5 Describing the Design Concept (Stage V)

In a very real sense, all of the early design described to this point is conceptual design. None of the previous description, however, has sufficiently emphasized the need for early design to eventuate in a preliminary, but comprehensive and integrated, conceptual model of the proposed system. This model must pull together all of the results from earlier determinations and analyses to define the instrumentalities and functions by which objectives will be achieved and constraints circumvented.

The design concept description, then, is not a productive design stage in the sense of moving system design beyond the allocation of functions. Rather, it utilizes all of the products and activities of previous design stages and incorporates them in a comprehensive system model. Since describing the design concept does not involve a set of design procedures, the

purpose here is to identify the essential components which should be included in the description. The design concept does not need to be in excruciating detail, but it must be a solid technical answer to the question, "what system to achieve what ends?"

The design concept that identifies the system and the ends it is to achieve integrates the following design products. The notation in parentheses identifies the textual source of each:

1. Selected development area (6.1.A)

2. Development rationale (6.1.A)

3. General requirements and objectives (6.1.B)

4. User operations (6.1.C)

5. System requirements and objectives (6.1.D, 6.2.C, 6.4.C)

6. Set of resources and constraints (6.2.C, 6.4.C)

7. Functions description of the system: process statements, mission analysis, function breakouts, dependencies and subsystems, and system models (6.3.B)

8. Functions allocation (6.4.C)

6.6 Determining Design Feasibility (Stage VI)

The feasibility study comes as a companion to the design concept description in terminating the early design effort. The feasibility question comes in two major parts. The first has to do with how well the system design concept meets all of the operational realities it would encounter if actually carried through the operational stage. The second part of the question has to do with development feasibility. This second aspect is probably best answered in terms of a concrete plan for the remaining stages of development, including schedules, manpower-facility-equipment requirements, monetary costs, and technical approaches.

It is the former feasibility question that largely concerns the design team. Determining design concept feasibility is an essential intermediate activity falling between pencil-and-paper design studies and the actual development engineering and production of system components. In a limited fashion, this design phase tests and checks out critical interfaces, operations, and system component configurations against the operational environment. The actual feasibility studies are obviously system specific, but generalizable information and tools can be derived from <u>Data Methods</u> and <u>Design Assessment</u>.

It then becomes the responsibility of whoever controls the developmental pursestrings to decide whether or not the system will move to the detail design stage.

CHAPTER 7

DESIGN ENGINEERING - HARDWARE

This chapter examines the nature and extent of hardware considerations in system design. The process of matching equipment or equipment specifications to allocated hardware functions describes, or at least affects, your job as a system designer. The primary concern of this chapter, then, is to identify the factors and activities involved in generating system hardware. Hardware design procedures are collectively termed Design Engineering and are divided into three design and development stages--detailing the design (Stage VII), engineering development (Stage VIII), and producing the system (Stage IX). An overview of Design Engineering activities is presented in Figure 7-1.

The technology of circuit design or other detailed engineering functions involve a level of specificity inappropriate to the broad perspective of hardware considerations described here. This approach will be of greater value to you than lengthy exposition of technical detail. The hardware component of system development is treated in a conceptual fashion and characterized as a generalizable progression of activities which results in the required system equipment. Hardware design and development breaks out from hardware-oriented considerations of early design once functions are allocated to men, machines, and programs. It encompasses the equipment configuration design and analysis efforts which permit final management make-or-buy decisions, testing and evaluation of hardware elements, and their production. The broad levels of hardware design activities are graphically illustrated in Figure 7-2.

A practical and detailed approach to implementing, integrating, and evaluating specific hardware capabilities in two sample information systems is outlined in Appendix 2, TRACE. TRACE identifies the series of technical tasks through two system analysis efforts which generate required system equipment lists. TRACE serves as a working example of the process by which the hardware elements and the design and development concepts outlined in this chapter are implemented in specific information systems.

7.0 Hardware Elements

Hardware, by virtue of its technical and tangible nature, is a far less elusive concept than software. While hardware has evolved by generations, its function in information and all automated systems has remained relatively constant. For purposes of this handbook, hardware refers to the physical and permanent components of an information system. This equipment is composed of engineered units or elements through which data are collected, reduced, extracted, communicated, processed, stored, viewed, measured, recorded, reproduced, converted, or calculated.

It is useful to adopt a hardware structure classification to distinguish among the equipment elements. Figures 7-3 and 7-4 illustrate the main classifications of hardware within current information systems. The central data processing hardware depicted in Figure 7-3 is generally the most complex portion of the total system. However, systems design and development efforts must also specify a set of peripheral hardware in addition to the central data processing hardware as shown in Figure 7-4.

Technology and environment greatly affect the point of departure for the design and development of all subsystems: hardware, software, and personnel. By evaluating hardware technology and the system environment, it is possible to define the status of each hardware component described in this section. Since this handbook is primarily oriented toward information systems which employ central data processing hardware, particular emphasis is placed upon identifying central data processing components that should be considered in a system design and development effort. The chapter summarizes the following classes of hardware which are integrated in the total system:

1. Central data processing components.
2. Communication components.
3. Hard copy storage and retrieval components.
4. Presentation components.
5. Measurement components.

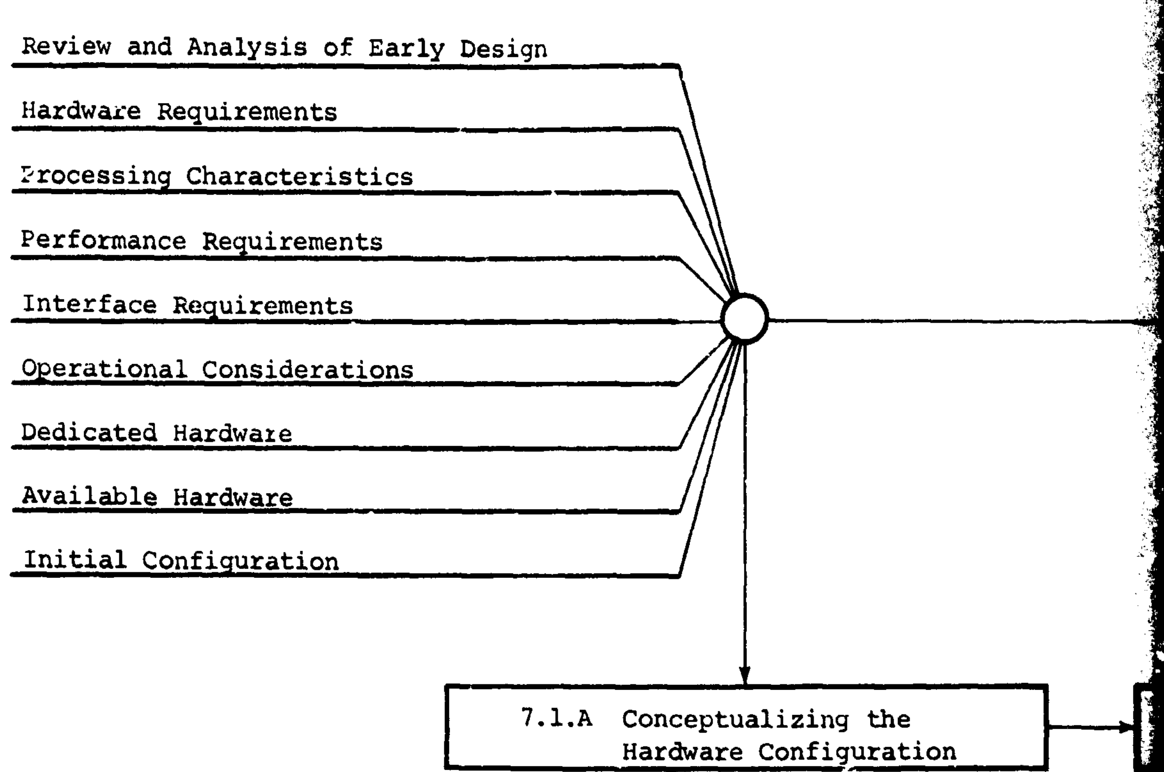
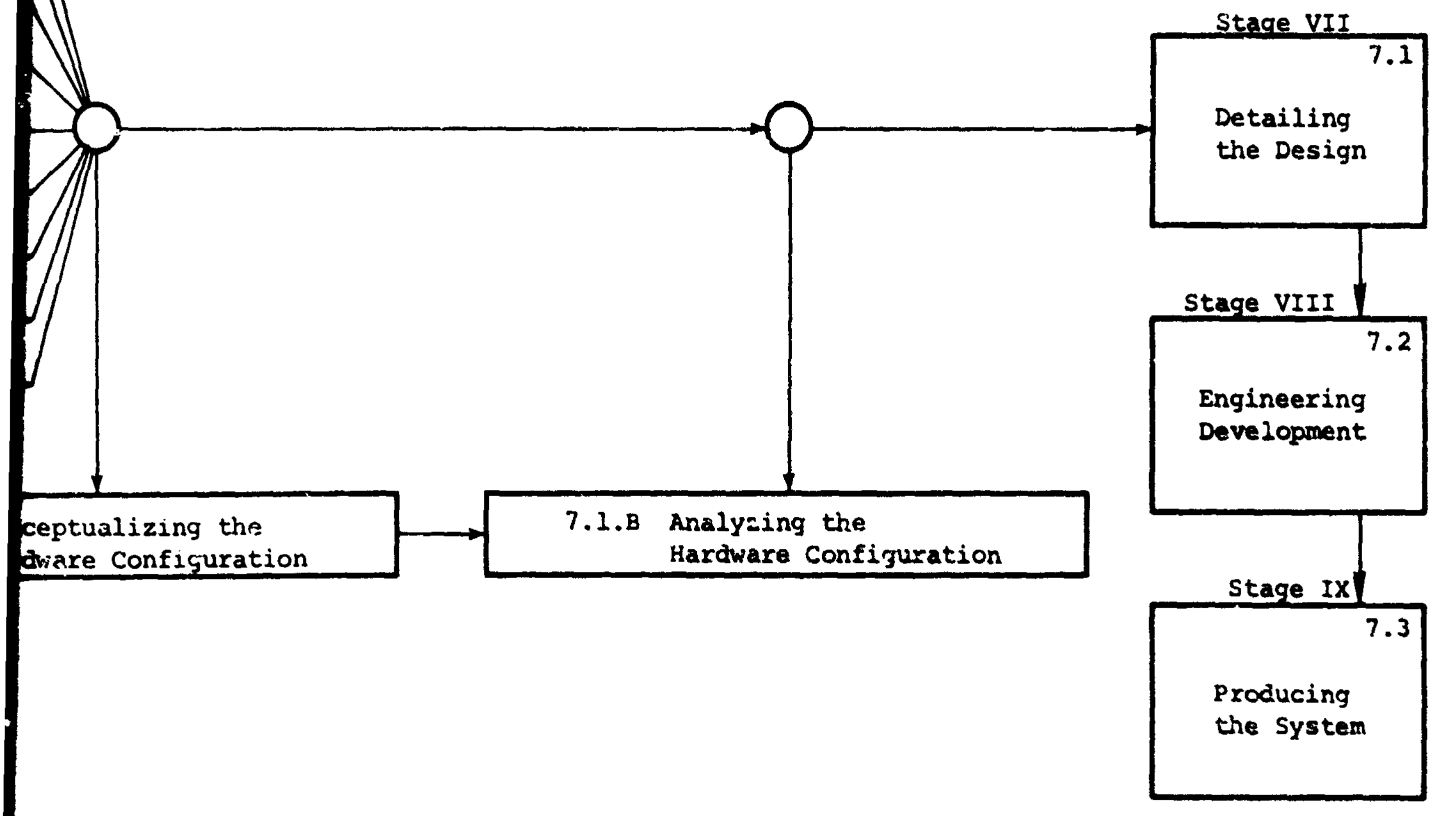


Figure 7-1. Overview of Hardware Design Engineering Procedures

H



ering Procedures

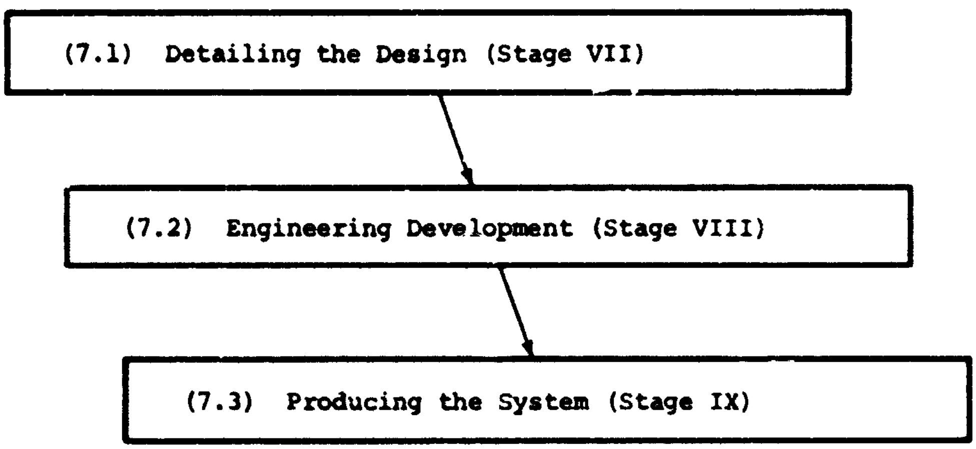


Figure 7-2. Stages of Design Engineering

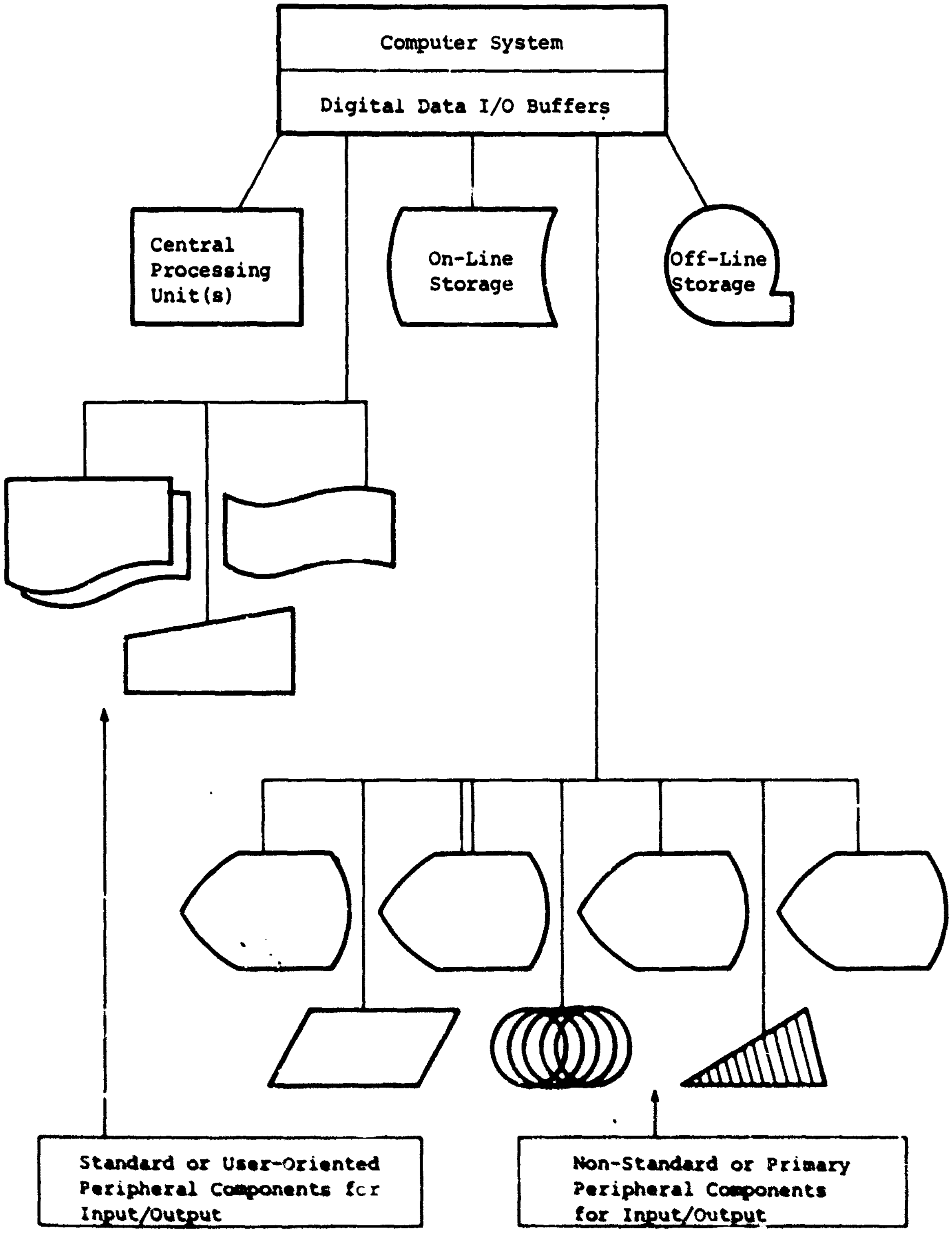


Figure 7-3. Central Data Processing Hardware

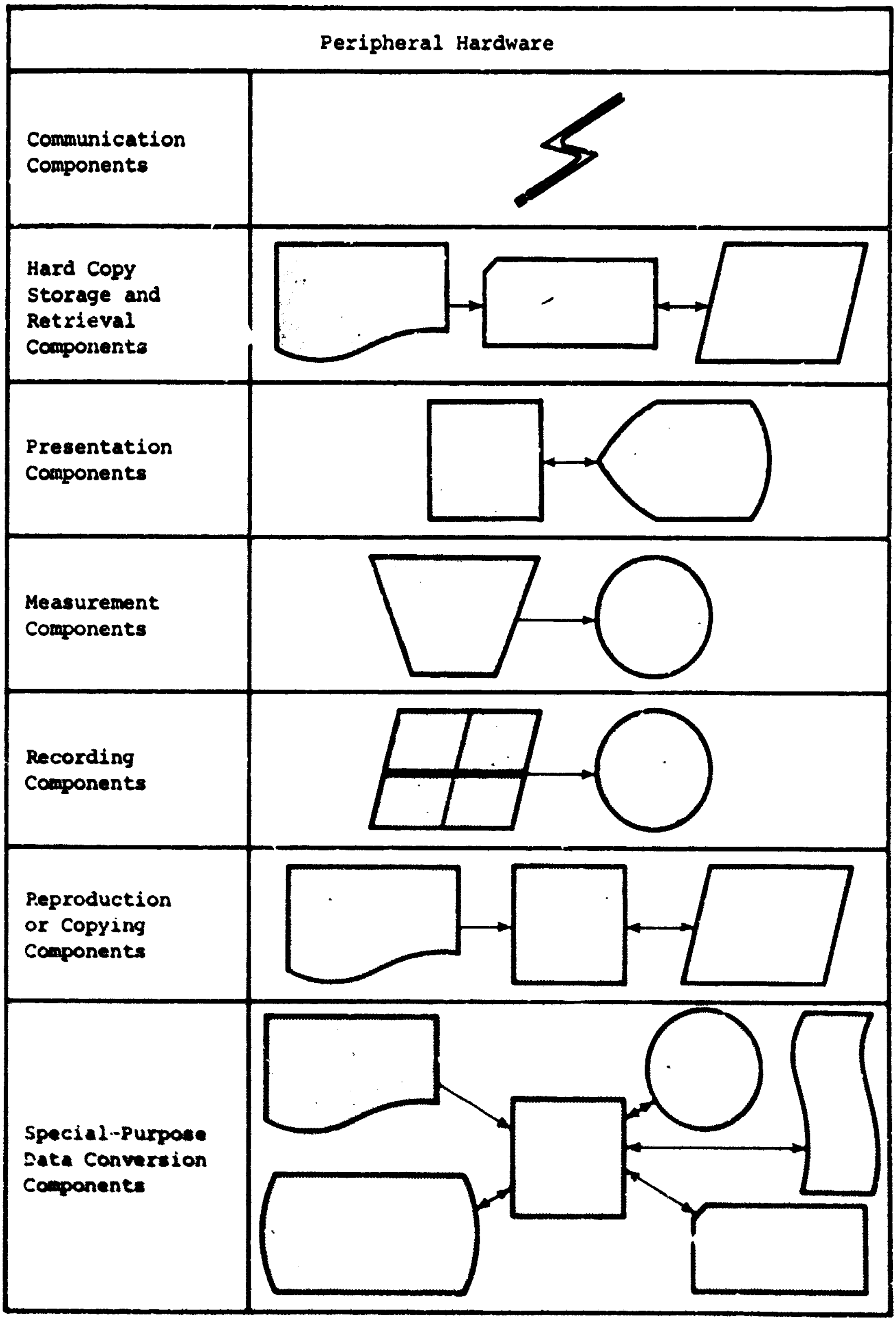


Figure 7-4. Peripheral Hardware

6. Recording components.

7. Reproduction or copying components.

8. Special purpose data conversion components.

Central Data Processing Components

Central Processing Elements. The central processor consists of four major units:

1. Arithmetic unit--performs numerical computations and logical comparisons in problem data.

2. Control unit--by means of switching circuits, directs the flow of information through the system and automatically times and sequences the necessary operations on the data.

3. Memory unit--stores information before, during, and after the processing of that data. Memory units also store the program sequence of instructions which direct the computer to perform the required information processing. Memories used in central processors are configured for the following internal functions:

    a. Discrete single bit storage for logical functions.

    b. Individual one word or partial word registers for mechanizing control registers, index registers, and arithmetic registers.

    c. Fast multiple-word addressable storage of limited capacity for control memory and scratch-pad purposes.

    d. Read-mostly storage for micro-programmed systems.

    e. Large-capacity high-speed internal data storage.

    f. Large-capacity high-speed internal program storage.

g. Buffers and speed-matching registers.

The types of internal memories normally encountered within central data processors are of the following types:

a. Integrated circuit memories.

b. Tunnel diode memories.

c. Planar magnetic thin-film memories.

d. Cylindrical magnetic thin film (plated wire) memories.

e. The permalloy sheet toroid memory.

f. The laminated ferrite memory.

g. The flute memory.

h. Magnetic core memories.

i. Delay line memories.

j. The woven screen memory.

k. Continuous-sheet cryogenic memories.

l. Ferro-acoustic memories.

4. Input/output unit--transfers information into or out of a computer based on the receipt of interrupts. It is the interface between the electrical world outside the computer which activates peripheral equipment and that abstract mathematical and logic arrangement of electrical signals inside a computer. Sometimes this area is further organized into a communication controller or input/output buffer unit if the data being handled is voluminous enough to warrant such an approach.

Mass Storage or Data Transfer Elements. Prerequisite to the successful development of any data processing system is the ability to store data in a form that allows data transfer between one piece of equipment and another.

Such transfer of data frequently occur over a considerable distance, a lapse of time, or after the intervention of some form of human control. Three types of data storage and transfer media are summarized in this section. These are:

1. Unit record elements (e.g., punched cards, magnetic cards).
2. Incremental record elements (e.g., paper tape, incremental megnetic tape).
3. Block record elements (e.g., magnetic tape, disks, drums, and off-line solid stage storage units).

Magnetic disks, drums, and transportable core or film storage units may also be used for on line storage. The more common mass storage or data transfer equipment is described below:

1. Card punches. Common to all punched card punches are the following elements:

   a. The feeding mechanism.

   b. The punch mechanism.

   c. The stacking mechanism.

   In addition to these three basic elements, many punches include a pre-read station before the punch, a post-read station following the punch, and a reject hopper.

2. Card readers. The elements provided in a punched card reader are similar in most respects to those provided in a punched card punch. In fact, in more modern equipment, it may be difficult at first to detect the difference unless the equipment is actually operating. The punched card reader consists of three basic elements:

   a. The feeding mechanism.

   b. The read mechanism.

   c. The stacking mechanism.

Like the punched card punch, a post read station and one or more additional stacking mechanisms are frequently provided.

3. Magnetic tape transports. Regardless of the central or random access memory size available to a computer system, there exists a requirement for storing data for record purposes. This "record data" is information that must be available to be reentered into a computer system at some later point in time. It is desirable, therefore, that such data is in computer language, recorded at a rate compatible with the speed of the recording computer, and available for re-entry at a rate compatible with the computer which later uses the data. Key parts of any magnetic tape transport element are:

    a. Start-stop mechanisms of which there are pinch-rollers, vacuum or pressure capstans, or clutch capstan types.

    b. Tape buffing mechanisms such as vacuum column, mechanical tension arm, or tape bin types.

    c. Recording and reading mechanisms which vary in technique as to the number of tracks and the method of recording timing control information.

4. Paper tape reader/punches. These are used primarily as computer input/output devices, and create the various kinds of punched tapes which are elements of mass storage. Some units are both reader/punches while others have the reader separate from the punch portion. The units are generally designed around the kind of punched tape to be handled as follows:

    a. Oiled paper tape: paper tape lightly and uniformly impregnated with oil for lubrication and easy punching.

b. Nonoiled (dry) paper tape: a nonoiled, more common, less messy paper tape (some tape punches require oiled paper, some tape readers require nonoiled paper tape).

c. Mylar tape: a tape with a plastic base for durability, uniformity, and strength. It may be blackened, aluminized, or both.

d. Chadless paper tape: a paper tape in which data are represented by partially cut holes, leaving flaps attached to paper ribbon and folded back. This paper tape is easy to punch, and needs no chip basket. However, it cannot be read by some tape readers.

e. Chadded paper tape: in this tape holes are fully punched out, requiring collection of chips (chad); it can be stored in less space, owing to reduced thickness (no folded-back flaps).

f. 5-, 7-, 8-level tape: there are many encoding schemes parity-type error-detecting codes may or may not be used; channel 8 may be reserved for an end-of-record signal or not be reserved.

In addition, the physical form of punched tape may be any of the following, depending on length:

a. Strips: usually 2 to 4 feet long.

b. Fan-folded: usually longer than strips, but shorter than rolls or reels.

c. Rolls: up to 700 feet long, usually read from center of roll, not outside end first. This saves rewinding when reading data in the same sequence as when punched.

d. <u>Reels</u>: up to 1000 feet long, usually read from the outside end.

Two principal hole-punching formats are now recognized:

a. <u>Center-feed</u>: the sprocket holes are centered on the same centerline as each character of data (10 per inch).

b. <u>Advance-feed</u>: the sprocket holes are not in line with the data, but somewhat ahead. This tape cannot be read on some readers. Advance-feed allows visual identification of the beginning end of the paper tape.

5. <u>Magnetic disk units</u>. Magnetic disks read and write on flat, circular plates with one or two magnetizable surfaces, on which data can be read or written by magnetic recording techniques. The disk is made to rotate about its center at high speed (like a very fast phonograph record). Data are recorded in circular data areas called tracks. Data are read or written by moving a read/write head to the track position while the disk is spinning.

The recording technology is similar to that of ordinary tape recording, with one important difference: the heads do not touch the surfaces, but float or fly about one-thousandth of an inch away. This flying-head construction reduces the attainable data density somewhat, but provides essentially zero wear of the record surfaces or heads, with a consequent very high data reliability. In modern usage, there is usually one head per recording surface, mounted on a comblike access mechanism. Indeed, in some disk drives there are four or more heads per surface. Each additional read/write head adds to the cost, but reduces access time and optimizes other design parameters. In the designs that have one or more heads per surface, the recordable area is

viewed logically as being divided into cylinders. A cylinder is all the area that can be used in one position of the access arm. This results in a series of concentric magnetic drums rather than parallel plane surfaces. This view is adopted because track-to-track transition is accomplished electronically within a cylinder, but mechanically and therefore slowly between cylinders.

6. Magnetic card units. These mechanisms handle cards with a magnetic surface on which data can be stored by selective magnetization. The card is usually made of durable but flexible plastic material and coated on one side with a mixture of magnetic oxide particles in a suitable binder. The entire construction thus resembles a piece of an uncommonly wide (and thick) magnetic recording tape.

   Information is recorded on the card in tracks (longitudinal narrow strips) each of which contains many hundreds of bits of information. Along each track the magnetic material is fully magnetized (saturated) in one direction or the opposite direction. The location and existence of each magnetic flux reversal serve to encode information on the surface. Information is read from or written on the card by mechanically moving the card past fixed read/write heads similar to those used in conventional tape recorders.

   By using multiple (individually removable) magnetic-card bins, a random-access storage of almost any total capacity can be constructed. These devices are characterized by large capacity, low cost, and slow speed.

7. Magnetic drum units. These comprise units having cylinders with a magnetizable external surface on which data can be read or written by magnetic recording techniques. Drums are made to revolve at high speed about their axes while many read/write heads float a few millionths of an inch off their external surfaces. These surfaces are usually

plated with magnetic alloy and very highly polished. The surface of a drum is divided into circular tracks, each as wide as a read/write head. In most devices the heads remain fixed, and each defines a track. There exist some movable-head drums with a behavior similar to that of magnetic disks.

Drums offer large capacity data storage with the fastest access of any mechanical (moving) storage, but at relatively high cost per digit because of the need for many costly read/write heads and for precision machining of surfaces.

8. Special purpose storage units. Optical disks, magnetic ink in cards, and text to be read by optical character readers are examples of other mass storage techniques considered in certain instances for special data storage applications. These items overlap into peripheral input/output areas and are often categorized as such.

User-Oriented Peripheral Elements. Peripherals receive information from and transmit information to the user environment. Input-oriented devices code operational data into a prescribed form and record the available input media. The output-oriented units accept the problem or process solution in the form of electrical pulses, arrange the pulses into significant character groupings, and transmit them to the user environment. Cathode-ray tube displays, teletypes, digitizers, and X-Y plotters are examples of user-oriented peripherals. Some of the more significant items in this component area are as described below. Since the peripheral elements of central data processing hardware within any information system are of major importance, it is appropriate to present a more comprehensive overview of these hardware elements.

1. Input-oriented hardware elements.

   a. Speech recognition. In theory, voice input represents a practical means of computer input as speech is the easiest technique which

man has of expressing himself. However, the variety of expressions complicates the problem of designing useful voice recognition equipment. A great deal of research has been performed that could eventually lead to a useful voice input system, including frequency analysis techniques and matrix comparison techniques as well as electronic, optical, mechanical, and digital analysis approaches. In spite of some limited successes using highly constrained monosyllabic vocabularies, there is little likelihood of widespread applicable voice recognition equipment being available for use in the near future (5-10 years).

b. Keyboards. Keyboards are designed to enter alphanumeric and symbolic information. These numbers and letters are usually supplemented with other non-spoken symbology, such as %, and punctuation marks. Keyboards are numeric, alphabetic, symbolic, or any combination thereof and can be designed to meet many special needs. Although many non-standard keyboards are available for special purposes, there are three standard keyboards that are generally accepted: the alphanumeric or typewriter keyboard, the numeric ten key keyboard, and the numeric bank or columnar keyboard. While many variations occur within each of these standards, there is enough stan dardization to allow the training of personnel in their operation.

Alphanumeric keyboards are designed to operate at a peak repetition rate for a single character of ten or fifteen times per second. As most alphanumeric keyboards are not interlocked to prevent

the simultaneous depression of characters, it is possible to operate such keyboards at speeds up to 20 characters per second providing that the same character is not repeated in sequence. Typical operator rates are about five characters per second when copying from legible data.

Ten-key numeric keyboards are designed to be operated by one hand. A trained operator can produce output at the rate of ten to twenty characters per second for reasonably long periods of time.

The bank or columnar keyboard provides a column for each digit position. Each column contains all of the digits which may be entered in that position, usually 1 through 9. This keyboard can be used as a fixed format entry device as all zeros or all blanks are automatically entered in each column when no key is depressed.

c. <u>Function switches</u>. Function switches are a form of selection device used to indicate change or initiate a machine action. Function switches are employed singly or in groups such that the selection of one function switch from one group modifies a selection of other switches from another group.

One of the biggest problems in the application of function switches is their number and location. Since each switch represents an idea or "concept" communication to the computer, there is usually not enough space available on the console to allow expression of all of the ideas that must be communicated. One approach which

has been used with considerable success by designers of command and control consoles is to produce a matrix of switches, each of which generates a unique code. This matrix is covered by an overlay which identifies the function of each switch within the matrix. The matrix overlay is itself coded in a manner such that the computer can sense which overlay is being used, and by first sensing the overlay in use and then sensing the switch being depressed, can tell the function to be performed. In this manner, a 10 x 20 matrix of switches with 100 overlays is used to provide unique identification of 20,000 separate functions. The disadvantage of such a system is that it takes an excessive amount of time to sort out the correct overlay and position it.

Another approach to the problem is to allow the computer to generate a series of labeled boxes or points on a display followed by operator selection of one of these with a light pen or other position indicator. In this manner, the compuer keeps the operator continually informed from which switches it is capable of accepting information. Further, if there are a large number of "overlays" in use, the computer can display a number or description for each overlay to facilitate operator selections.

d. Graphic input devices. Graphic inputs such as light pens and joy sticks provide a type of input device for geographic and geometric data. They can be used to designate a point or line, the parameters of which are already known to the computer, or they can be used to draw graphical

input which can be reduced by the system into useable data. A wide variety of position indicators and graphic input devices are now in use, and most depend on one of three basic techniques. These techniques are potentiometers and gray-scale coding, scanning, and matrix sensing. Each of these techniques is able to indicate a single point to provide a coded X-Y coordinate for computer input, to select a point from a number of computer generated points to indicate a single point, and to generate a line or series of continuous points.

Potentiometers and gray-scale coding technology is used in devices such as the telautograph, joy sticks, ball indicators, and pantographs. The underlying technique of these systems is to couple the pen, ball, or joy stick to an X axis indicator and Y axis indicator. Where analog output is acceptable, these X and Y axis indicators are a pair of linear or rotary potentiometers. As the indicator is moved, the resistances on the X and Y potentiometers vary as a function of the position of the indicator. When digital output is required, a linear or rotary gray-scale indicator is used in place of the potentiometer. The gray-scale indicator is a sensing device which allows a position to be read as a direct binary code.

Gray-scale coding is equally useful for both point selection and line drawing. As the resolution of this type of system is dependent upon the fineness of gray-scale provided, a line drawn between two points will always generate

the same number of X, Y coordinates regardless of the speed with which the line indicating instrument travels.

The most commonly used scanning device is the light pen. The basic technique used in all photo scanning devices is to place a photo sensitive receptor (the light pen) over a point on a phosphorescent screen. As the surface of the screen is scanned by an electron beam, the phosphor is activated momentarily and generates visible light. When the beam arrives at the point over which the light pen is resting, the phosphor is activated, generating visible light which is detected by the photo sensor in the end of the light pen. This generates a pulse which is transmitted back to the scanning system. The pulse is used to read out the X and Y beam deflections and thus indicate the position of the light pen.

Scanning may take place in a number of fashions including horizontal raster, circular raster, and point sequencing. Point sequencing consists of successively activating the phosphor at a number of target points, the locations of which the computer is constantly tracking. When the target indicated by the light pen is sequenced, it can thus be identified. Point sequencing allows the position of computer generated points to be sensed but does not provide for the entry of other graphic data.

Raster scanning techniques may be used with a light pen to produce line input; however, since the pen position is scanned at a fixed interval of time, the number of X, Y coordinates that can

be identified along a line will depend upon the speed with which the light pen traces that line. In most cases, the scanning rate (or point sampling rate) is high enough to provide reasonable line resolution.

e. Character recognition. Although a wide variety of character recognition equipment has been developed, most is special purpose. This equipment has potential application in all areas where a hard copy document is originated for human use prior to entry of data into the computer system. While typical applications include inventory control and personnel records, character recognition equipment has also been applied to reading an intelligence data base.

Currently, many techniques are available suitable for reading alphanumeric characters and other symbols from printed copy. Some use special type fonts while others use a variety of conventional type fonts. Among those suitable for use with limited fonts are: magnetic ink character recognition devices such as those used in the banking industry, stroke analysis techniques, split font techniques in which a code designating a character is carried in a series of fine breaks in the lines forming the character, and bar coding techniques where a code representing the character is carried in a series of fine bars that are printed immediately above or below the character.

Types of character recognition techniques suitable for reading one or more general purpose type fonts are:

1) Scanning techniques in which the character is broken into a series of horizontal or vertical scans, each of which contains a pulse train that can be analyzed in sequence.

2) Matrix techniques in which the character is broken into a series of small areas each of which may be compared for correlation against a master matrix.

3) Gross comparison techniques in which the total character is compared against a series of masks for coincidence, and corner detection techniques in which line intersections are detected and their number and relationships compared against a coincidence table.

All character recognition techniques are dependent upon being able to segregate the character to be recognized from the background noise of the paper. This noise results from the differences in coloration between various areas of the character, lack of definition of the edges of the character, voids, dirt, and ink deposits on the paper. A printed character is never perfect and can seldom meet all pre-established criteria necessary for recognition. Recognition is a probability problem in which probability of correct reading must be gauged against the probability of misreading the character. Therefore, character recognition equipment is designed to accept a character as valid when it meets a given percentage of total criteria. If the pattern of a

character cannot meet an adequate percentage of criteria when compared with each character possibility, it must be considered an unrecognizable character. As character recognition equipment becomes more discriminatory in accepting a character, a greater number of characters that cannot meet criteria standards occur; with the result that a higher number of characters cannot be read. In order to read more characters, it is necessary to lower the criteria necessary for character verification with the result that the error rate increases.

2. Output-oriented hardware elements.

   a. Printers. Two basic types of printers are available. They are the line printer and the character printer or mechanized typewriter. The line printer prints one line of data at a time--printing speed is dependent on the number of lines printed and independent of the number of characters printed per line or of the total number of characters printed. Such line printers can be divided in four classes according to their functional printing characteristics. These classes are:

      1) Electromechanical.

      2) Electro-optical.

      3) Electrographic.

      4) Magnetic.

   Character printers are designed to print one character at a time horizontally across a piece of paper. Printing speed is directly proportional to the number of characters and control

actions that must be taken by the printing device. The use of this type of device usually requires a number of special control functions and corresponding special control codes. Typical of these are: space, back space, precedence, e.g., upper and lower case. The character printer is usually used as a communications device, as a part of a document originating device, as output on a console, or as a very low speed output device. Character printers are electromechanical in nature and are, therefore, capable of producing carbon copies. All operate at speeds between ten and twenty characters per second and use alphanumeric type fonts. For purposes of detailed examination, electromechanical character printers can be divided into five classes:

1) Typewriters.

2) Drum printers.

3) Ball printers.

4) Matrix printers.

5) Stick printers.

b. <u>Plotters</u>. A wide variety of plotting techniques are now in use. Functionally, they include recording galvanometers, servo potentiometers, sweep recorders, pressure recorders, and X-Y recorders. The first four of these techniques are used primarily for the continuous recording of physical phenomena. Typically, they employ a moving roll of paper or film which passes under one or more "pens" which are activated

by a positioning device, e.g., galvanometer, servo, etc. Characteristic of all these techniques is that they plot variations in physical phenomena against a time base represented by the moving paper. Since the paper moves at a fixed speed in one direction, most are not capable of plotting phenomena that are not time based. Usually, these plotters are designed for direct coupling with some form of sensor. To conserve paper they are designed to operate at the slowest time base that allows observation of the maximum anticipated variation of the phenomena under study. Their sensitivity is usually determined by the output of the sensing device to which they must interface. Typical applications include the electrocardiograph, pressure recorders, voltage recorders, temperature recorders, vibration recorders, etc. This class of device is not generally suitable for use with a computer-oriented system.

X-Y recorders do have application in many information systems. Two types of devices, X-Y plotters and cathode-ray tube (CRT) camera recorders, are described below:

1) X-Y plotters. X-Y plotters are suitable for recording both time dependent and non-time dependent functions because the X and Y pen positions are independently controlled by two separate servomechanisms. Furthermore, X-Y plotters can operate in two modes; the point plotting mode

and the line plotting mode. In the point plotting mode, X and Y positions are furnished the servos to position the pen and the pen is depressed to record the plotted point. This process is repeated for each point plotted. In the line plotting mode, the pen is initially positioned as a point plot and is kept in contact with paper while the X and Y servos move the pen from point to point.

Both analog and digital X-Y recorders are available. The analog plotters use conventional servo motors while the digital plotters use digital stepping motors to position the pen. Accuracy and speed of X-Y plotting techniques is primarily dependent on the accuracy and speed of the digital stepping motors used to position the pen. Motor capacity required is a function of the inertia of the system which will vary with the size and mass of the pen positioning bars (a function of plot size) and whether the pen positioning bar must carry accessory equipment such as small character printers for point identification.

Two implementations of X-Y plotters are in common use; the table plotter and the drum plotter. The table plotter uses the large sheet of paper

which is anchored to a flat base. Over this base, horizontal and vertical plotting bars are moved to position a pen holder at this intersec ion. This pen holder is capable of placing the pen point in contact with the paper upon command. Some plotters use a small character pri⁚⁺.r in place of a pen to print a series of characters or points. Other systems employ the pen in combination with a character printer to imprint point identification next to the plotted points. Typical table plotters can provide pen movements up to 50 inches per second with positioning accuracies of 0.01% of the sheet size. Character identification can be added at rates up to 10 characters per second. When table plotters are used in a vertical position as a group display, it is necessary to sacrifice some plotting speed and accuracy because of gravitational effects on the vertical trace.

Drum plotters represent another implementation of the X-Y plotter technique. In the drum plotter configuration, the paper is wrapped around the drum which is rotated by a digital stepping motor to produce the Y axis position. The X axis position is produced by moving the pen back and forth on a fixed carrier ba.. Because of the

mass of the drum, plotting speeds are somewhat slower than can be achieved with table plotters; typical speeds being about three inches per second. Two major advantages of the drum plotter technique are that it requires less space than the table plotter and that the length of the Y axis used is limited only by the amount of paper that is supplied to the plotter. A major disadvantage of drum plotters is that only a limited portion of the plot is available for examination during the plotting process and this is in motion, making plot reading rather difficult.

2) Cathode-ray tube camera recorders. The cathode-ray tube camera recorder is a high speed equivalent to the X-Y plotter. Horizontal and vertical deflection plates are used in a cathode-ray tube for the same function that servo positioners or digital stepping motors are used in X-Y plotters. The electron beam generated by the cathode-ray tube gun is moved across the phosphorescent screen on the face of the cathode ray tube. Depending on the phosphor used, cathode-ray tube plotters may be used for direct view or projection, e.g., displays, or in conjunction with photo-sensitive paper or

film for a recorded output. This type of plotter is capable of very high speeds; however, it is not too suitable for point plotting because of the persistency of the phosphor necessary to convert the electron beam into visible light. The major differences between cathode-ray tube plotters and displays lie in the phosphors used on the cathode-ray tube and the availability of a photo sensitive copying technique to produce a hard copy.

c. <u>Vocal output</u>. In the past few years there has been a rapid rise in the development of equipment which is capable of selecting a prerecorded audio message and presenting it upon a digital command. There has also been some work on equipment that is capable of selecting a variety of words and phrases and combining them into a meaningful message. As yet, there has been little such equipment put "on line" with a computer system. This equipment has manifested itself as automatic paging systems and selectable message systems, etc.

d. <u>Cathode-ray tube displays</u>. Display hardware is typically designed for specific systems so that consoles vary in capability and capacity. Nevertheless, console characteristics currently appear to be more uniform. Large group displays have never been produced in any quantity so that it is almost impossible to speak of their typical features. It is possible, however,

to classify display hardware characteristics and from such a classification develop system building blocks. Furthermore, such an approach permits the development of a software concept and operating system.

1) Individual CRT displays. The following features are common to most CRT type displays:

   a) Alphanumeric Keyboard. This consists of a set of keys comparable to a standard typewriter keyboard. In addition to the letters and figures, punctuation and special symbols are included. Sixty-four possible characters are usually (some reserved for control functions) available, since 6 bits are conventionally used for symbol r resentation. Since there are 43 keys on a typewriter, this implies the need for a shift key or augmented keyboard.

   b) CRT. This is a cathode-ray tube or oscilloscope unit capable of displaying a set of characters or symbols with line drawing as a possible option. Generally there is a one-to-one correspondence between the symbols available on the

alphanumeric keyboard and those that can be generated with the CRT. Tube face sizes are usually 19 to 21 inches.

c) Function keys. This set of keys is assigned to application-oriented procedures. Individual keys may represent a call for an action, or groups of keys may be associated, forming a message calling for action. These keys are usually under program control. To make the device general purpose or multi-purpose, it is desirable to have the meaning of these keys vary on demand of the operator. One convenient way to achieve this variability is by a replaceable mask or overlay as incorporated in several commercial products.

d) Status indicators. Data processing system status, both internal computer and console, is shown by status indicators. These indicators may be labeled neons; their "on" or "off" conditions indicate status.

e) Alarm indicators. Alarms or error indications are conveyed

by a set of labeled neons. Buttons may be associated with these lights for operator recognition and resetting. Audible alarms may also be included.

f) Control keys. These keys are assigned to specified tasks and support functions by which system control, data entry, and status requests are made. These keys are usually under hardware control.

g) Light pen. The user/operator can index any symbol on the CRT by using a hand-held photoelectric device.

2) Group displays. Group displays are still in their infancy and have not satisfactorily proven their utility except possibly for stylized displays in military command and control/operations. The important user-oriented characteristics distinguishing these devices are:

a) Alphanumeric readout. Typically 64 character representation is possible. Variations in size may be an option. Since often the image is generated by the CRT (available in a console system), the capability of the console display

can be duplicated with the group display (except for such dynamic capabilities as blinking of characters).

1) Vector drawing. Two very different technologies are employed in vector or line drawing. They are related to the nature of producing the display itself. One approach, typified by film based systems, will produce the display in its final form having all lines completed. A second approach, represented by systems employing a scriber etching a path on a plate, produces the line while the display is viewed.

c) Color. The question of color, and how many colors, is a fundamental consideration. Typical choices include a black and white system or a system employing the three primary colors from which it is then possible to obtain mixtures which will afford eight colors, including black and white. An interesting question is whether or not the color

must be true or whether it is only necessary to achieve sufficient discrimination capability to facilitate communication of information. Simplification in hardware and cost differentials are possible if the latter capability is adequate.

d) Projection overlays. This is an important feature found on many group displays at present. Maps or grids are projected from a fixed set of slides for use as overlays with the computer generated and displayed data.

e) Screen size. The size of the group display viewing area is dictated by the number of required viewers. Popular screen sizes are 4 x 6 feet or 8 x 10 feet. Another parameter to consider is the question of how many screens and the possible requirement of a simultaneous or coupled screen display.

f) Projection techniques. Possible alternatives are front and rear view projection. Selection of one or the other is a function of the physical layout of the presentation room.

g) <u>Library feature</u>. Display systems which save and store displays have a library capability. Film based systems typically have this feature when slides are saved for later recall. The implication of the library feature in this case is that it may be necessary to have an off-line library capability with attendant bookkeeping and searching. Display recall may also be implemented by requiring the computer system to remember "old" displays; when they are referenced, the computer will regenerate them.

h) <u>Display control</u>. A control panel from which selection and/or requests can be made must be available to the display users. A simpler control unit can be used to access information held in the library. In this case a dial-up process might be sufficient, in which the display of a specified magazine slot or film position is requested.

i) <u>Alternate hard copy</u>. This is usually a function of the information system and is normally

implemented through a printer device. It is a hardware function in that a hard copy film record is produced by the group display at the time of presentation.

j) High ambient light viewing. This is a most desirable feature in a command area where written messages and discussions are extensively utilized.

Key Operational Criteria. Central data processing components can be described in terms of several broad attributes:

1. Sensitivity--The hardware must be capable of detecting environmental signals relevant to the system. This includes the ability to minimize acceptance of signals not intended to be signals and rejection of those which are in fact signals.

2. Transduction--The hardware must possess the ability to carry and convert the form of data as required within the system.

3. Capacity--The hardware must be devised to handle varying rates of data transmission and meet duration of performance requirements.

4. Compatability--The hardware must adequately respond to input from and output to other system components or incorporate sufficient buffering devices.

5. Reliability--The hardware must be capable of repetitive and accurate responses to data input and consistent handling of data output.

6. Flexibility--The hardware must be so designed that it can be altered or expanded to meet changing requirements without major interruption of system requirements.

7. Maintainability--The hardware must be devised to facilitate maintenance operations primarily of the preventative type and encompassing corrective maintenance considerations.

Communication Components

Communication components are the physical means of connecting remote locations so as to transmit and receive information. Communication links such as telephone lines, microwave relays, teletype lines, and coaxial cables have great importance in information systems with the advent of time-sharing and real-time systems. Four examples of the many types of linking that involve computers are shown in Figure 7-5.

Systems considered for military applications generally involve communications interfaces. The communication aspect is particularly important when classified information is transmitted and especially when multiple levels of classified data are handled. Advances in communication techniques such as the implementation of broad bandwidth channels via satellite and image transmission by employing analog to digital to analog conversion equipment create additional interfaces which can interact in information systems. Message switching technology and encryption technology discussions are outside the scope of this handbook. However, the point here is that the communication components of any system must be identified early in system analysis, and the equipment or procedural interfaces clearly understood by the time a system specification is prepared. There often are hardware, software, personnel, support, and facility elements that relate to communication components within large, complex information systems. As a result, it is important to have an experienced communications-oriented team member involved in the early stages of system design efforts. It is extremely detrimental to system implementation if significant communication interfaces prevent data utility from an information system. That is, the users who must obtain information

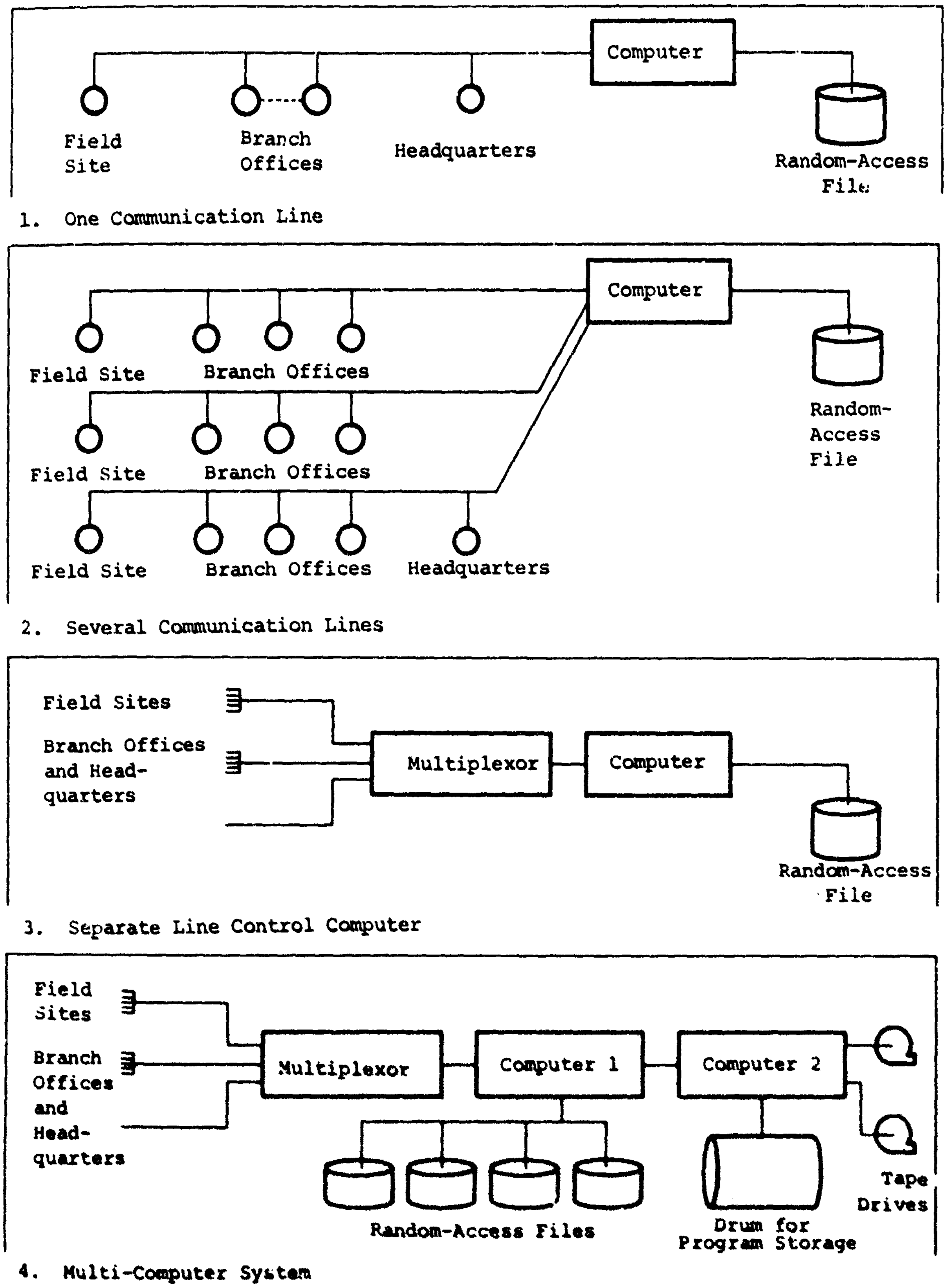


Figure 7-5. Computer Communication Links

do not receive the system output. The data output as well as the data input communication interfaces of an information system must be examined before system acceptance is ever complete.

Hard Copy Storage and Retrieval Components

Many information systems are required to store and retrieve hard copy items containing a variety of kinds of information. Photography maps, textual reports, and micro-film copies in unit record or roll transparency formats are examples of the hard copy items typically considered in system design. A key decision in any information system is to determine which information will be in hard copy formats versus digital formats. Then, within the hard copy portion, it is necessary to specify the breakdown of specific formats such as photo transparent graphics or text, opaque graphics, miniaturized copies, rolls or films or paper, unit record forms, etc. Some of the commonly identified hard copy storage and retrieval components are as follows:

1. Roll film storage units.
2. Unit record storage units.
3. Viewers for miniaturized transparent records.
4. Automatic indexing and retrieval query consoles for roll and unit record items.
5. Special purpose copying equipment that format hard copy products for further storage and retrieval.
6. Automatic routing or transmission devices for disseminating the information on hard copy storage media; i.e., air tubes or closed-circuit TV equipment.

These components are very often a part of information systems and frequently interact with the central data processing components in indexing and query-related functions.

Presentation Components

Many information systems are designed in support of operations which convert or create data extracted by personnel viewing some form of analog record, i.e., graphic, textual, photographic, or actual physical objects. Some of this presentation equipment is placed in the CRT display category; however, the items of interest here are those hardware elements that are more optical-mechanical than electronic. The types of equipment that often must be analyzed or considered in this category are:

1. Film projectors.
2. Monocular magnifiers.
3. Stereoscopes.
4. Image enhancement devices.
5. Light tables.
6. Status board displays.
7. Reflecting projectors.

Some presentation elements perform storage and retrieval functions or are tied to system measurement components. They often dictate, however, a significant portion of the data input characteristics and output requirements for information systems. In addition, presentation elements generally comprise an important portion of the information system itself.

Measurement Components

Analog formatted data very often must be quantized in order to determine the dimensions, weight, volume, or quantity of particles as key inputs to information systems. Measurements are frequently tied directly to central data processing components or to man-machine input techniques employed as substitutes. In either case, the equipment characteristics of required measuring aids are important if measurements are performed as a part of an information system. Examples of the key types of measuring devices frequently involved in system analysis and design are as follows:

1. X-Y measuring aids for determining distances on imagery and graphics.
2. Angular measuring aids for determining point relationships.
3. Area and volume measuring aids that combine linear and/or angular values for size determination.
4. Weight measuring aids for size or growth determination.
5. Particle content measurements for solution status determinations or density calculations.
6. Speed and acceleration measurements for various applications.
7. Direction measurements for various applications.
8. Time measurements for change or status information.
9. Miscellaneous energy measurements (light, heat, electrical values) for various applications.
10. Point count measurements for sampling or other data analysis applications.

Measurement components are often outside the information system to be developed. They are, however, important contributors to data requirements if they interface with system input. An early task in system design and analysis is to specify measurement elements if they are at all involved in the system.

Recording Components

There are many types of data recording hardware elements involved in information systems. While information systems commonly interface with recording components, these components may actually be included in the system by management direction or operational necessity. The types of recording elements considered in system design are as follows:

1. Sound recorders.

2. Visual and infrared energy spectrum recorders.

3. Mechanically activated data recorders for man-machine input operations.

4. Recorders interfaced with measurement components.

Dependent on their application, these hardware elements are frequently included in the user-oriented peripheral area of central data processing components, as part of communication components, as part of measurement components, as part of reproduction or copying components, or may be considered as special purpose data converters. The frame of reference and experience of the system design staff dictates how recording components are labeled. The important point is that these hardware items often affect information systems to some degree and, if so, should be considered throughout all phases of system development.

Reproduction or Copying Components

Reproduction or copying hardware elements are those items that use optical-mechanical-electrical techniques in combination to reproduce analog, hard copy information which facilitates physical data dissemination or manual storage. Examples of this class of hardware are:

1. Photo-copy enlargers.

2. Photo-copy contact printers.

3. Photo-copy reduction printers.

4. Electrostatic text and graphic copiers.

5. Xerographic text and graphic copiers.

6. Offset printing equipments.

7. Thermoplastic copy equipment.

8. Chemical-sensitive reproduction equipment.

Reproduction or copying hardware is primarly related to personnel, support, and facility [illegible] 4. In general, software is minimally related to

this particular class of equipment. Reproduction or copying equipment does, however, significantly affect input and output data requirements in many system analysis efforts.

Special Purpose Data Conversion Components

This category of hardware elements is very dependent on the orientation of particular system users, management policies, and the technical background of the system design team. The types of equipment considered to be special purpose data converters are those elements especially designed to convert analog data to digital data and vice versa. They are configured to handle special purpose data involved in specific limited applications. Examples of conversion equipment are as follows:

1. Color scanners for digitizing cartographic graphics.
2. High resolution scanners for digitizing imagery and reconstructing the imagery.
3. Electronic image restitutors for changing the relative perspective of points in a photograph.
4. Optical character readers.
5. Automatic densitometers.

Again, special purpose data conversion elements are identified as such by system developers and users for convenience and understanding, coordination of system functions or element relationships, and communication to individuals or groups with different backgrounds. Many items placed in this category can just as logically be listed in other hardware component categories previously described. The point here is that hardware elements should be identified as special purpose data converters when advantageous to a system analysis effort. If specified early in design activities, these hardware elements will be properly installed as needed by the particular system. As design and development progresses, special purpose data conversion elements can be placed into more appropriate hardware categories.

7.1 Detailing the Design (Stage VII)

The previous section presented the scope of hardware elements generally considered in information systems. The initial design activities related to these hardware elements are described in the system design and development stage termed detailing the design. Hardware design comprehends a series of complex information exchanges, technical analysis, and specification modifications. It is characteristically a refinement process. The manner in which the design activities are related and the essential characteristics of each are described in the following pages. Figure 7-6 identifies the major activities involved in detailing the design.

7.1.A Conceptualizing the Hardware Configuration

The process of formulating the overall hardware configuration requires several iterations before hardware selection/development is finalized. While available hardware as well as software solutions to system requirements facilitate the design task, hardware configuration is dictated wherever possible by desired operational capabilities rather than limited by ready-made solutions. However, knowledge of equipment capabilities and characteristics is certainly an essential requirement for design activities. The following possibilities are combined to eventually produce an optimum hardware configuration:

1. Modification of present equipment.
2. Replacement of present equipment.
3. Acquisition of new/additional equipment.
   a. Purchase off-the-shelf.
   b. Develop new system concepts.
4. Modification of design parameters to work within present equipment or feasible equipment modifications.

The multiple factors considered in hardware configuration and specification are weighted according to system objectives and requirements and the

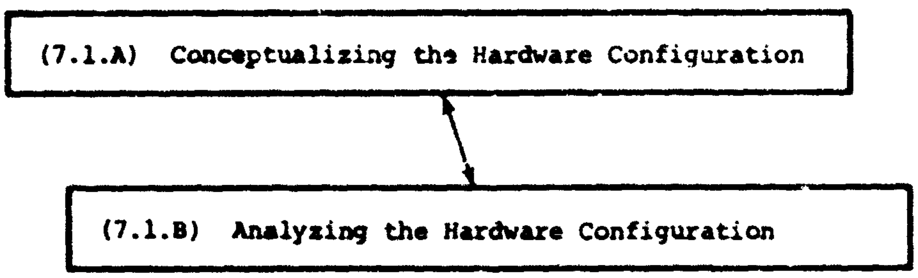


Figure 7-6. (7.1) Detailing the Design (Stage VII)

system functions allocated to hardware. It is assumed that efforts to conceptualize the process of hardware configuration herein will be supported with technical aids (i.e., System Computer Evaluation Review Technique [SCERT]) and resources available to the designer.

The generalizable design activities of the hardware configuration conceptualization are illustrated in Figure 7-7. A primary consideration from the onset must be the informational areas described in the review of early design. These system descriptors are the basis for evaluation and specification of hardware requirements.

Review and Analysis of Early Design. The initiation of hardware design considerations occurs in the integrated system effort of early design. The equipment elements of the system are designated as a subsystem with the allocation of system functions to hardware, personnel, or software. At this stage of design, the system concept is described in gross terms. Hardware subsystem efforts must ensure that a fully expandable equipment configuration is designed to meet the parameters or processing requirements of the proposed system. The preliminary attention of the subsystem must, then, be focused upon the system design concept of early design.

The object of the review of early design is to assess the direction of hardware-oriented activities and determine their fidelity to system requirements. Early design provides the source of reference for future hardware design efforts as well. Description of system performance requirements and system goals are immediate input to hardware development. Since functions allocations are still tentative at this point, it is necessary to review allocation decisions to ensure that hardware elements possess effective and adequate processing capabilities. Possible hardware-software and, to a lesser extent, hardware-personnel tradeoffs are relevant design considerations.

Hardware Requirements. The specification of performance requirements for functions allocated to the hardware subsystem first involves identifying how the data processing equipment must operate in the system environment. The hardware functions are derived from the system concept formalized in early design. Generally, the functions must be examined in terms of input and output characteristics:

1. Purpose.

2. Location, remote or central.

3. Devices utilized to perform functions in the past.

4. Characteristic descriptions:

    a. Range of values.

    b. Format.

    c. Frequency.

    d. Response time requirements.

    e. Source or goal.

It is possible, upon extensive examination of the input and output characteristics of the functions, to determine the nature of the processing action required to transform the inputs into outputs and speculate about the hardware components capable of performing such transformations.

In an organizational context, the stated requirements should have the following characteristics:

1. Identify the relationship of the central processing unit to storage, input/output, and communication requirements.

2. State the necessary processor capabilities in terms of:

    a. Accessibility.

    b. Speed.

    c. Multi-programming or multi-processing.

In general, the specification of requirements involves translating gross system functions into equipment concepts. The process must be referenced to concurrent software and personnel activities to determine whether the parallel translation and specification activities are compatible.

Processing Characteristics. This activity is aimed at examining the processing environment of the system defined by the gross system concept. The mode of processing is defined within the first stages of the system

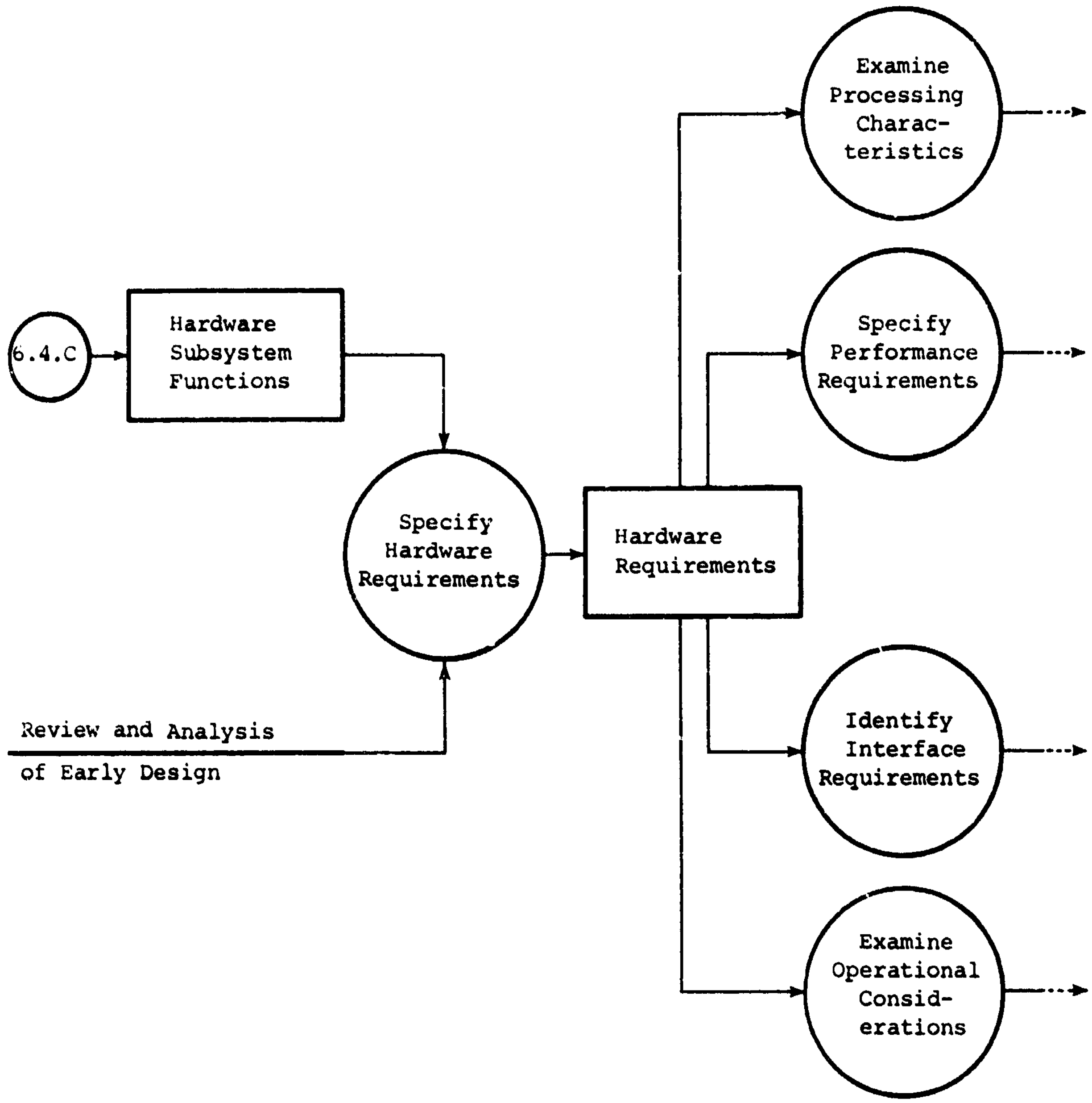


Figure 7-7. (7.1.A) Conceptualizing the Hardware Configuration

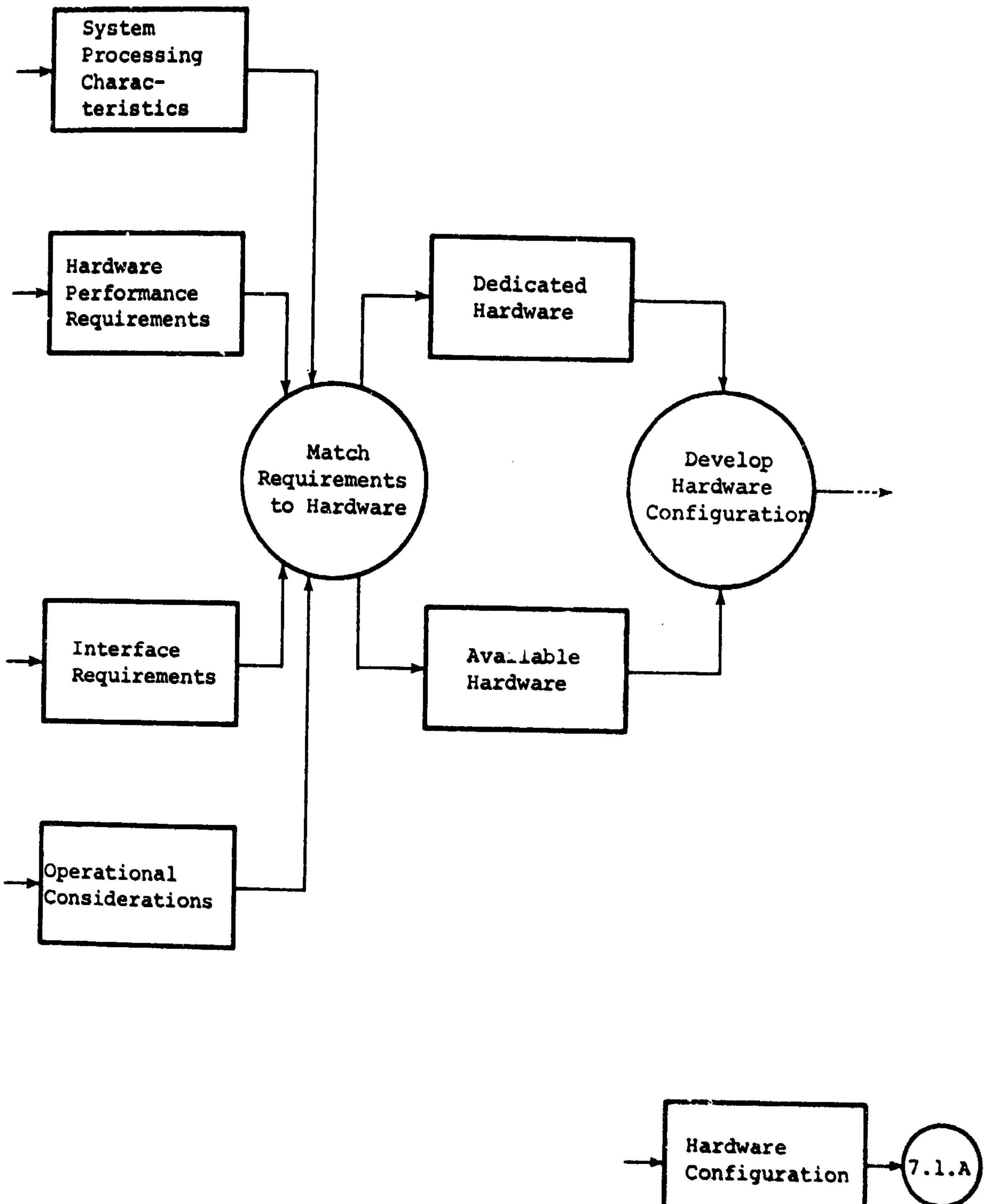


Figure 7-7. (Continued)

development effort and has significant implications for equipment specifications. The processing mode is generally batch- or communications-oriented, aspects of the former being incorporated in communications-oriented processing. The continuum of complexity, cost, and capability has as one extreme batch processing and as the other extreme communications-oriented processing. The latter extreme, generally required in information systems, consists of several types:

1. On-line real time systems with multi-programming and multi-processing capacity. Multi-programming refers to the time interleaved execution of two or more programs. Any device capable of performing more than one process at a time is a multi-processor--that is, a machine with either multiple arithmetic and logic units for simultaneous use or single arithmetic and logic units capable of multiple simultaneous instruction execution.

2. Remote entry with centralized control of batch processes.

    a. Collection of transactions at remote locations.

    b. Transmission to central processor-central control.

3. Remote entry with remote control of batch processes.

    a. Collection of transactions at remote locations.

    b. Remote control of central processor.

4. Integration of on-line and batch processes.

5. Time-sharing.

<u>Performance Requirements</u>. The hardware performance requirements are derived from hardware requirements and processing characteristics and describe the overall characteristics of the combined operation of hardware elements. While performance requirements are described initially in early design, the emphasis here is to detail the requirements in terms of a more technical context. The two major areas of performance characteristics are:

1. Accuracy. The minimum acceptable error rate for variations of system operation must be determined. Accuracy considerations fall within the realm of software and personnel components as well as hardware. The extent of hardware responsibility and the relationship to other subsystem controls should be specified.

2. Speed. The performance capability of the system hardware in response to the rate of transaction must be determined from function requirements. The minimum acceptable response time serves as a selection/design factor in developing the hardware subsystem.

The performance requirements in the above areas move with increased design and analysis activities from functional descriptions to engineering specifications, i.e., from a qualitative to a quantitative context. In general, the performance requirements serve as input to the analysis of the hardware configuration design task.

Interface Requirements. Hardware configuration and eventual selection must coordinate equipment requirements with personnel and software requirements. The interface consideration obviously includes equipment connections as well. In regard to personnel interface, equipment specification is referenced to human factors in the following areas:

1. Input/output equipment--display devices.

2. Console design.

3. Communication network equipment.

The system display element is perhaps one of the most important interface points since it is used here to refer to any presentation of data to people. Relevant display attributes related to equipment design are:

1. Visual presentation form.

2. Timeliness.

3. Accessibility.

4. Transduction mechanism.

5. Corrections.

Analysis of the interaction between the personnel and hardware subsystems precedes any final equipment selection or development. Human factors considerations are necessarily incorporated into the hardware design specifications.

As is emphasized in the next chapter, Software, software availability and capability are probably two of the most influential factors in accessing vendor-supplied hardware. The nature of software package considerations in relationship to equipment and system configuration is described in Software. Software is particularly relevant in meeting system processing requirements and in implementing the system hardware as well as software elements (installation software). The hardware subsystem design effort should consider the following software package features in particular:

1. Size of system programs--the amount of storage required for resident operating system segments.

2. Location of software segments--the tradeoff between throughput speed and storage media expense.

The equipment interface generally involves matching present equipment with new equipment and tying together new equipment elements. The required electrical connections are the province of electrical engineers. The feasibility of the interface, however, must be determined prior to equipment acquisition.

Operational Considerations. Hardware configuration comprehends a number of operational considerations which are relevant design parameters.

Fallback provisions. The extent of fallback capacity is dependent upon the critical nature of the system environment. Consideration must be given, however, to some degree of failure detection and recovery, and to the nature of the recovery. In general, fallback procedures are necessitated by the following four categories of emergency:

1. Failure of input or output units.

2. Failure of a storage unit.

3. Failure of a communication line device.

4. Failure of the main computer.

The hardware requirements in these areas must be specified. When automated recovery is an essential system characteristic, the equipment components should possess the following features which are integrated with software elements.

1. Error indication efficiency.

2. Automatic interrupt capacity.

3. Equipment release and reset.

4. Error condition storage and return to interrupt point capacity.

5. Automatic check of error detection circuitry.

6. Error logging equipment.

7. Memory-storage protection.

8. Device redundancy requirement.

Security techniques. Security measures in the system may or may not be a critical design consideration. Where the nature of the information processing is highly classified, the hardware components have implications for data security. A description of hardware security techniques, involving equipment features and added security features, is beyond the scope of this chapter; however, design considerations in this area may be referenced to the following technical report:

> Bingham, H. W. Security techniques for EDP of multilevel classified information. Griffiss Air Force Base, New York: Rome Air Development Center, 1965. (RADC-TR-65-415)

Expansion capacity. Anticipation of growth requirements is an early design consideration which must be examined in terms of hardware

component compatibility and software-hardware compatibility. The projected hardware expansion of the system utilizes state-of-the-art information concerning hardware and software technical expertise to evaluate the ease and cost of expansion. If it is feasible to incorporate hardware components with compatible instructional sets, then, little software modification is required in upgrading the hardware facility. In the event that direct transferability between components cannot be achieved, evaluation of emulation possibilities is a relevant consideration. If simulation meets the system expansion requirements, then the nature of the simulation should be examined.

Dedicated Hardware. The preliminary specification of hardware function requirements permits tentative identification of ideal equipment capabilities and configuration. However, realistic system development requires the incorporation of dedicated system components even though theoretically speaking, the design of equipment should be system-specific. The degree to which equipment presently assigned for system applications or already determined for use can accommodate the hardware requirements must be identified. This involves the following general activities:

1. Acquire equipment specification documentation.

2. Match equipment specification with functional requirement specifications.

3. Identify the extent to which functional requirements are met by dedicated equipment.

4. Specify modification requirements.

Available Hardware. Vendor-supplied hardware is the primary and often the sole source of new hardware components in the system. The identification of available hardware requires painstaking examination of hardware "pieces" or systems which have the capability to meet the processing requirements assigned to the hardware subsystem. The primary objective is, of course, to identify the hardware component combination which meets performance requirements most cost/effectively.

Initial Configuration. The hardware configuration conceptualization has the following essential characteristics, in increasing levels of technical detail as design efforts progress:

1. Describes applications in terms of required hardware components.
2. Describes the utilization of dedicated hardware components in the system.
3. States processor configuration preferences and capacities.
4. Identifies the total hardware subsystem performance requirements.
5. Describes communication requirements in terms of hardware components.

The hardware configuration conceptualization is generally integrated into an overall system description for vendor proposal action. Hardware system requirement specification in sufficient and accurate detail is necessary to enable vendors to produce realistic and appropriate system proposals.

7.1.B Analyzing the Hardware Configuration

The analysis phase actually occurs simultaneously with the hardware conceptualization process and is illustrated in Figure 7-8. The emphasis here as with all system design efforts is upon cost/effectiveness. Cost/effectiveness is assessed in terms of matching system requirements with the most efficient hardware element at the least cost. There are several technical considerations which may be examined to determine if the configuration is both efficient and effective in meeting system requirements. These factors are as follows:

1. Core storage utilization.
2. Processing time utilization.
3. Peripheral storage space.
4. Channel utilization.

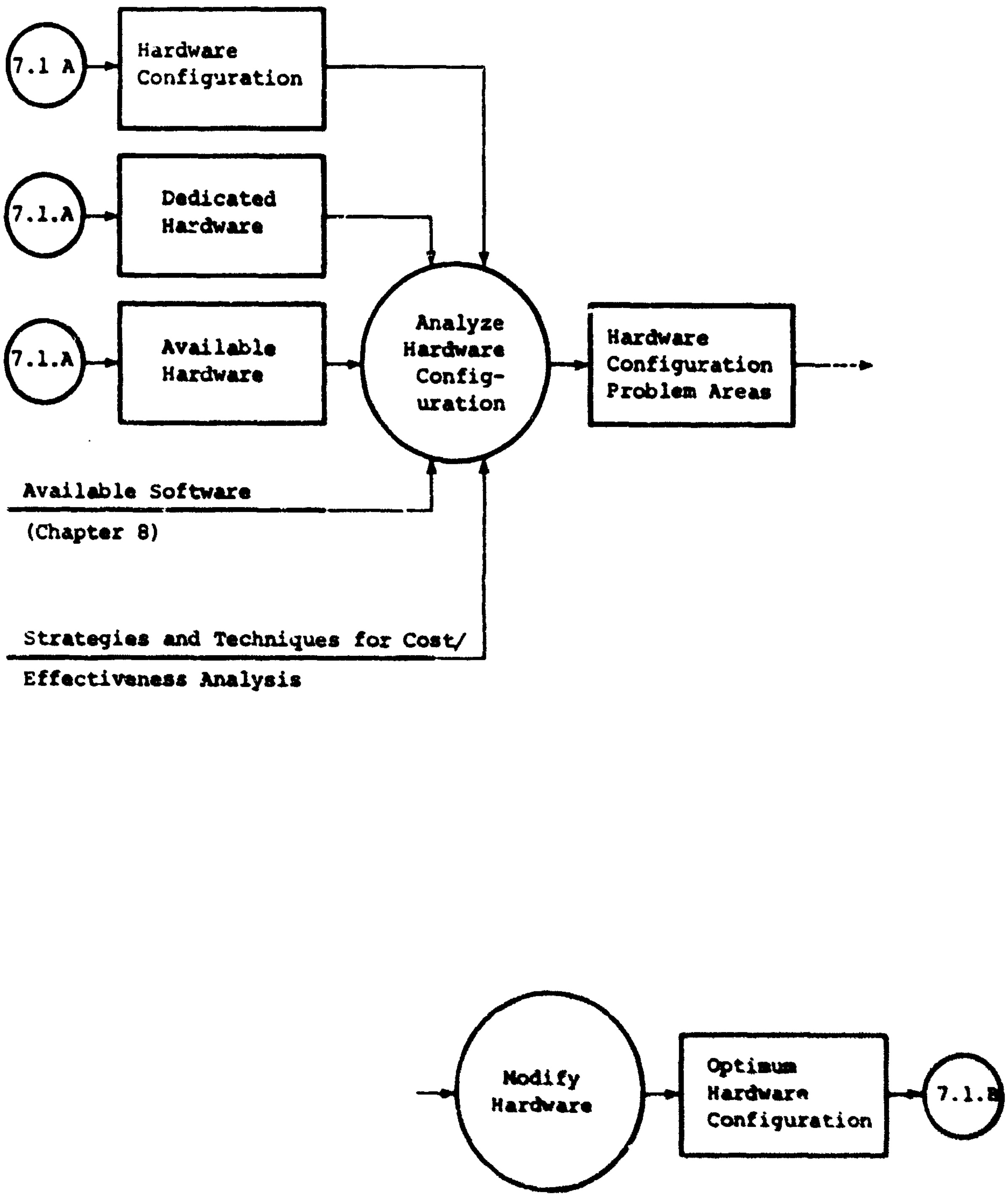


Figure 7-8. (7.1.B) Analyzing the Hardware Configuration

5. File arm or head utilization.

6. Line control utilization.

7. Communication line utilization.

8. Terminal utilization.

The specific techniques of cost/effectiveness analysis are described in a number of available sources. One convenient summary of these techniques is:

> ARINC Research Corporation. Guidebook for systems analysis/cost-effectiveness. Annapolis: Author, 1969. (AD 688 154)

7.2 Engineering Development (Stage VIII)

The importance of engineering development is to study the feasibility and implications of the hardware configuration in relationship to the system environment. Engineering development is characterized as a reconciliation process and is conceptualized in Figure 7-9. The hardware configuration is matched against presently available equipment, and tentative combinations of compatible items are considered until the best match is achieved. The following general considerations should be examined as a basis for eventual selection decisions.

1. Implications for total system design.

2. Initial cost considerations.

3. Efficiency analysis.

4. Implications for system performance and control.

5. Operating costs.

6. Additional equipment availability.

7. Personnel requirements.

The role of hardware subsystem development in relationship to parallel software development bears some consideration at this point. The relationship

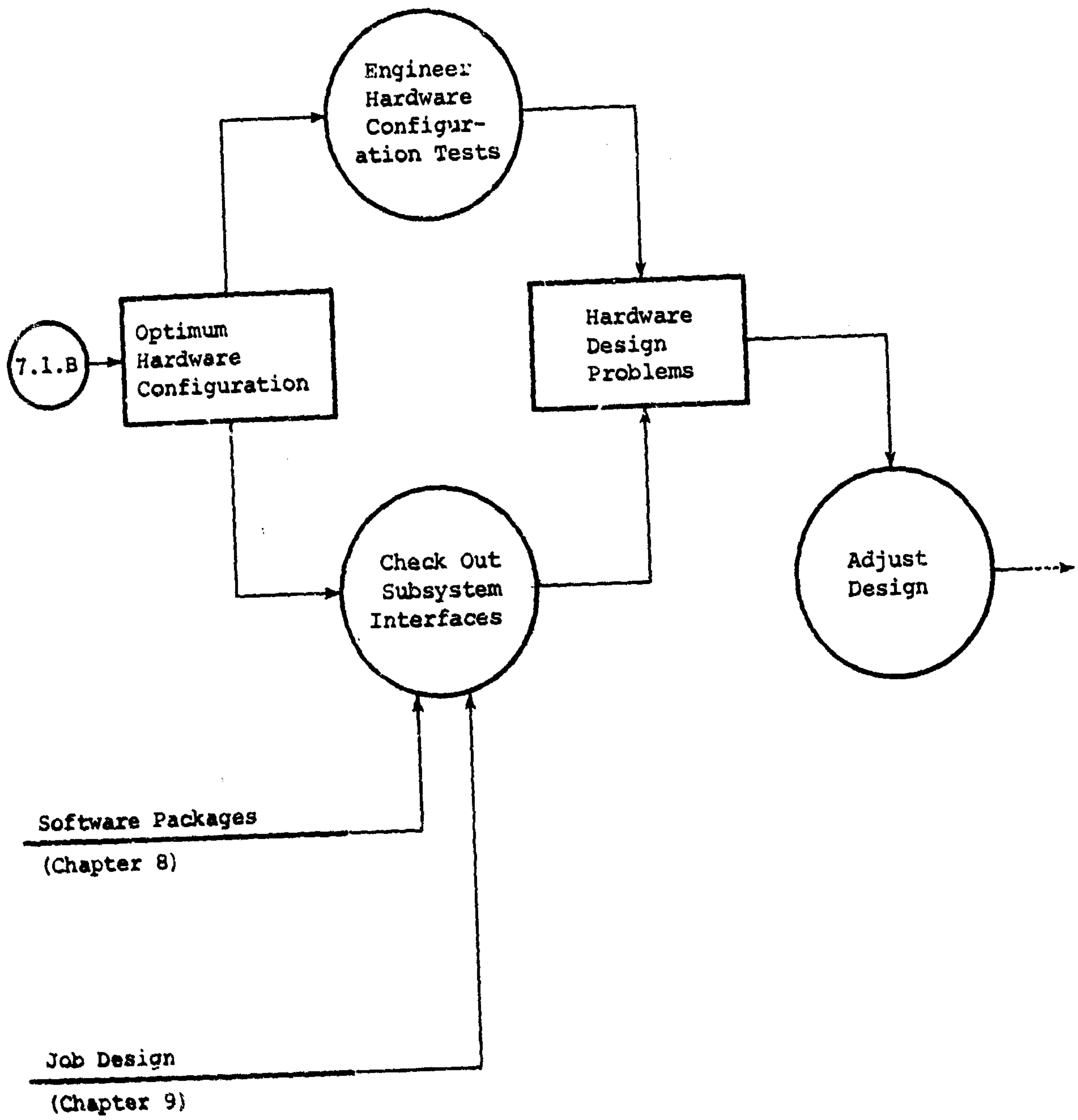


Figure 7-9. (7.2) Engineering Development (Stage VIII)

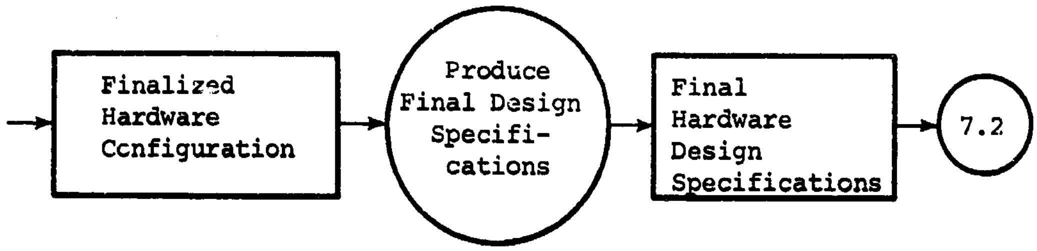


Figure 7-9. (Continued)

can be characterized by a few broad assumptions which refer to conventional general processor design. (Microprogram design is excluded since microprogramming is a functionally distinguishable approach).

1. Software conceptualization is bounded by hardware configuration.

2. Hardware configuration utilizes software availability as a resource/constraint parameter.

3. Hardware selection or detailed specification of equipment development operates as a given to program specification.

Therefore, it becomes apparent that in order for the conceptualization of software structure to be translated into design specifications, the hardware configuration must be virtually finalized.

7.3 Producing the System (Stage IX)

On a broad conceptual level, hardware production activities are illustrated in Figure 7-10. The initial evaluation of available hardware occurs in the conceptualization of the hardware configuration. At this level of the hardware subsystem development, however, the evaluation involves extensive analysis of vendor-supplied hardware proposals based upon system descriptions and the hardware configuration conceptualization phase which has proceded it. The following considerations are relevant in vendor-user hardware negotiations:

1. Personnel.
   a. Management.
   b. Technical support.
   c. Past performance.
2. Projected system.
   a. Cost.
   b. Performance.

c. Fidelity to requirements.

3. Maintenance considerations.

   a. Capability--the maintenance personnel qualifications and numbers.

   b. Availability--the location and organization of service units.

   c. Reliability--the response reliability based on performance history.

   d. Extent--the nature of the maintenance, encompassing preventative as well as failure support.

4. Installation factors.

   a. Timing--estimated time frame for delivery.

   b. Installation manpower support.

   c. Vendor-system environment.

      1) Equipment size and floor space requirements.

      2) Projected expansion space requirements.

      3) Electrical requirements.

      4) Temperature and humidity requirements.

      5) Equipment interface requirements and characteristics.

5. Testing considerations.

   a. System tests--to measure hardware capability in the system operational environment.

   b. Component tests--to measure the hardware component performance.

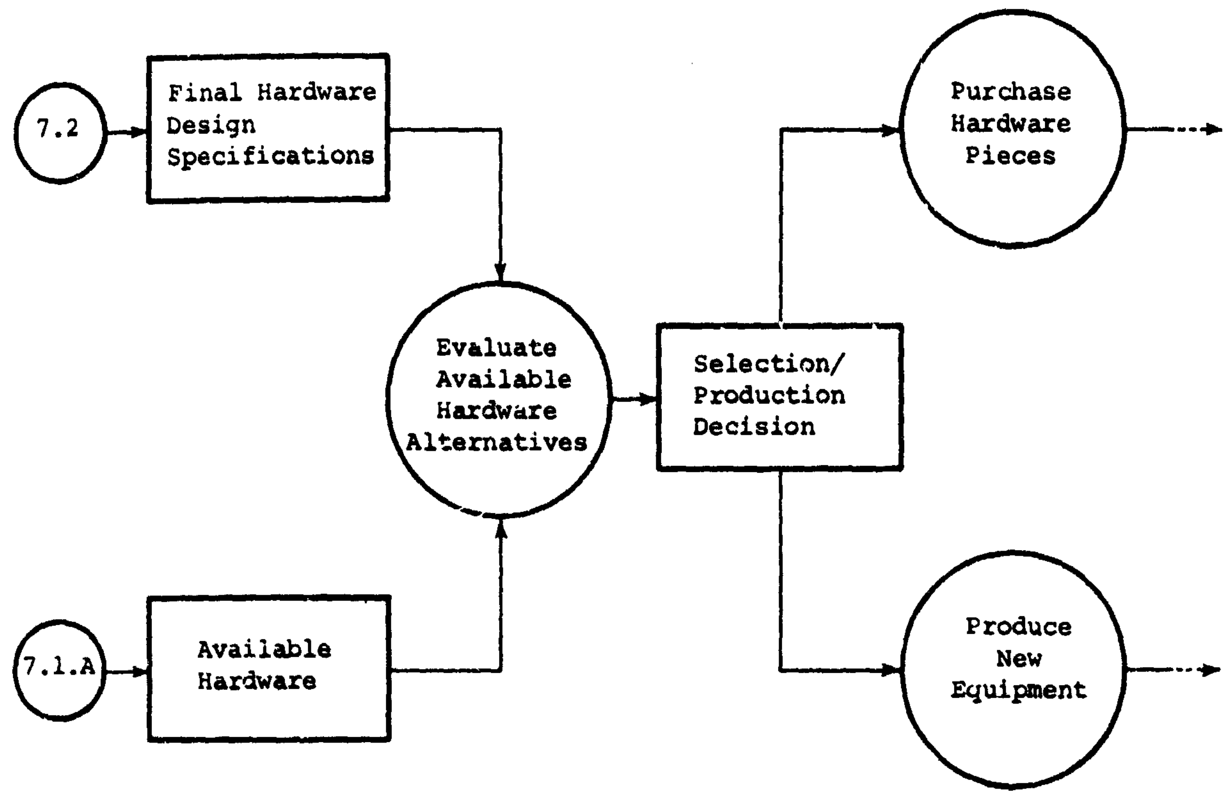


Figure 7-10. (7.3) Producing the System (Stage IX)

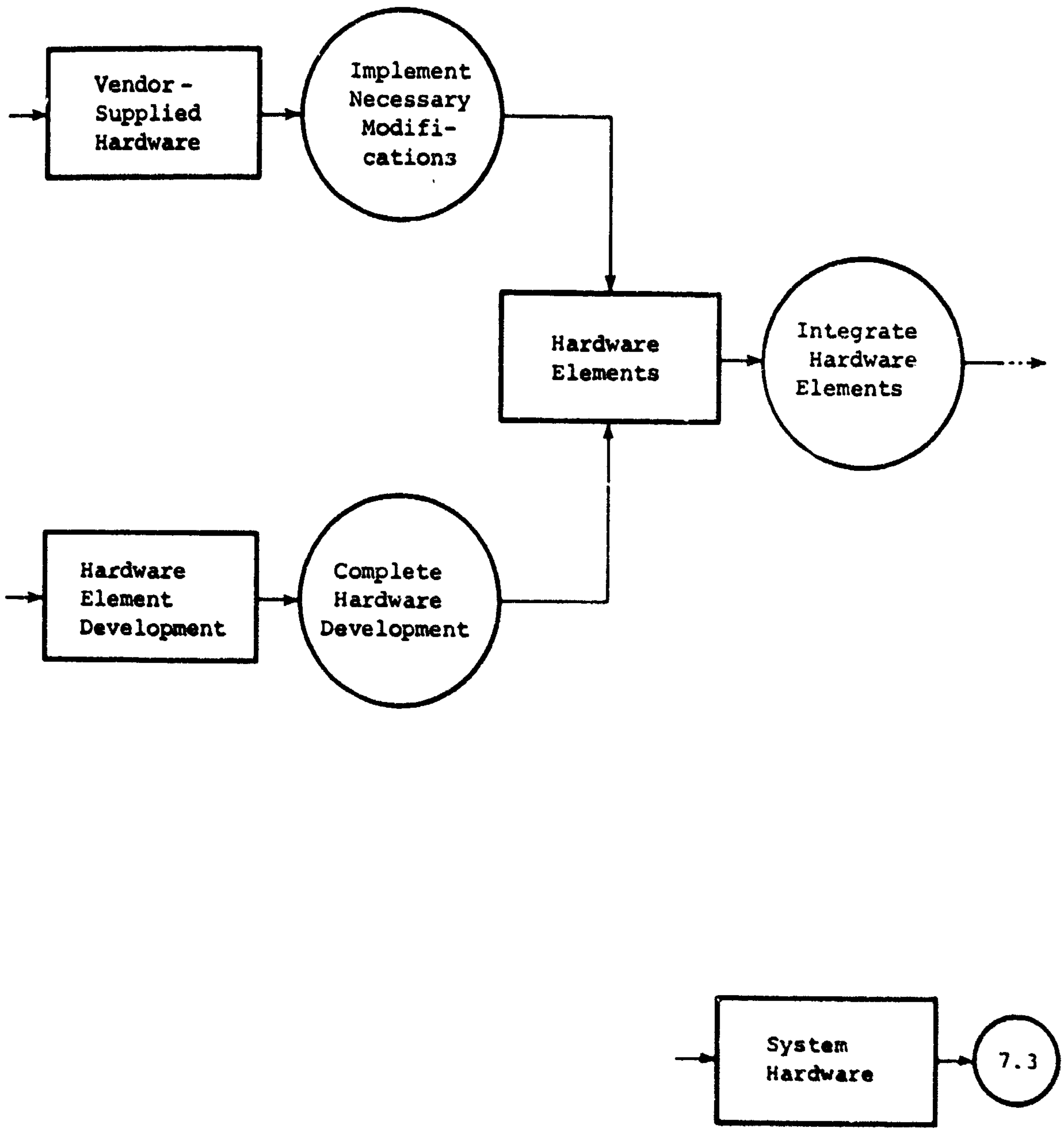


Figure 7-10. (Continued)

c. Program tests--availability of equipment or simulated equipment for program checkout and debugging.

In the final analysis the hardware can be acquired via vendor or can be produced specifically to meet the particular system requirements or can be a combination of the two alternatives. The phases of production differ to some extent with each alternative.

Purchase Available Hardware. The decision to purchase off-the-shelf hardware may require no equipment modifications or, more likely, may involve the alteration of certain hardware elements to meet system requirements. Assuming the latter condition to be the case, the following activities evolve from the purchase decision:

1. Determine the nature of any necessary component modifications.
2. Detail equipment specifications.
3. Compile detail design documentation.
4. Prepare drawings and other schematics.
5. Supervise vendor development and/or modification activities.

New Equipment Development. It is currently rare that anyone seriously considers development of central computing equipment for a specific information system. Digital computer development is almost always done as a major commercial venture for a general market. There remain, however, decisions concerning development of specialized peripheral devices and modifications of existing equipment to provide features not available from off-the-shelf equipment.

The decision to develop or contract to develop new equipment concepts is generally a major one having important cost implications and weighted by the projected performance capability of the new hardware elements versus available hardware. The nature of new equipment development activities is characterized below:

1. Detail design of new equipment.

2. Detail interface requirements of new equipment with:

    a. Present equipment.

    b. Equipment purchased off-the-shelf.

    c. Equipment modifications.

        1) Present equipment.

        2) Purchased equipment.

3. Compile detail design documentation.

4. Segment new equipment development activity.

    a. According to function.

    b. According to design requirements.

5. Prepare drawings and other sc matics.

6. Supervise new equipment development.

7. Bench, in-plant tests.

Integration. The production of hardware elements eventuates in the integration of the hardware subsystem elements for installation with software and personnel in a total system effort. Obviously, the subsystem activities are highly coordinated up to the point of system installation; however, the development efforts can be identified within subsystems. The integration of hardware elements is a priority consideration for the following reasons:

1. Checking out system programs and application programs.

2. Operator training.

3. Procedures che kout.

The final evaluation of hardware performance occurs in the operational shakedown of the system development effort. Appendix 2, TRACE, examines in detail the hardware subsystem in relation to two sample systems carried through all the tasks associated with system design and development to final hardware evaluation.

CHAPTER 8

DESIGN ENGINEERING - SOFTWARE

This chapter presents the nature and impact of software considerations in system development. It is a conceptualization of software development activities which identifies the essential characteristics of software acquisition in the system. Software design begins with the breakout of early design into subsystems and terminates with program production. Figure 8-1 is an overview of software Design Engineering, a term applied to the activities involved in software design and development. Design Engineering consists of three major stages--detailing the design (Stage VII), engineering development (Stage VIII), and producing the system (Stage IX)--which are identified in Figure 8-2.

The aim of this chapter is to facilitate your job as a systems analyst by creating a broad perspective of software development in relationship to total system development. The progression of activities which produce system software is described as a generalizable process. Since an abundance of literature treats the mechanics of programming activities, that aspect of software considerations is excluded from the chapter. Furthermore, the specifics of well-defined software areas such as file structures and design are referenced rather than described.

Appendix 2, TRACE, provides a detailed task by task approach to the application and development of software elements in two information systems. The practice-oriented format adopted in TRACE outlines the system analysis efforts which generate, implement, and integrate the software subsystem. TRACE serves as a working guide to the procedures by which the software elements and the design and development concepts presented in this chapter are carried out in actual information systems.

8.0 Software Elements

The preliminary focus of the chapter is upon establishing some definitional and conceptual boundaries of terms integrally related to software development. The definitional state of affairs in information systems is notably

poor--especially in the software area. The intention here is to establish definitions which will be used in a consistent manner.

For the purposes of this handbook, software refers to the totality of machine-manipulable programs and routines acquired (i.e., via vendor) or generated during the systems effort. Software development includes design, analysis, production/procurement, testing, and implementation as well as the documentation that accompanies these efforts. Effective software has the following characteristics:

1. Maximizes hardware efficiency.

2. Facilitates the tasks of program design and maintenance.

3. Directs the hardware in performing system-specific jobs or tasks.

4. Is consistent with and facilitates the performance of tasks required of personnel.

The software development process embodies two functionally distinguishable aspects--systems programming and applications programming. Certainly, the two are intrinsically interdependent, but the distinction is useful in terms of conceptualizing the software development process.

Systems programming embodies the first two software characteristics and refers to programs which control and guide computer operations, assist other programs and programmers with supporting functions, and increase the usefulness of the hardware. Included in this category is a variety of overlapping and occasionally synonomous program classifications such as utility programs, operating systems, supervisors, monitors, executives, service programs, support programs, application libraries, diagnostic routines, processing programs, and control programs.

The principal product of the systems programming effort is the operating system. Operating system here refers to the totality of programs which are indispensible to the total system operation in a control and support (subsidiary) capacity. An operating system has the following functions, dependent upon the extent of the system:

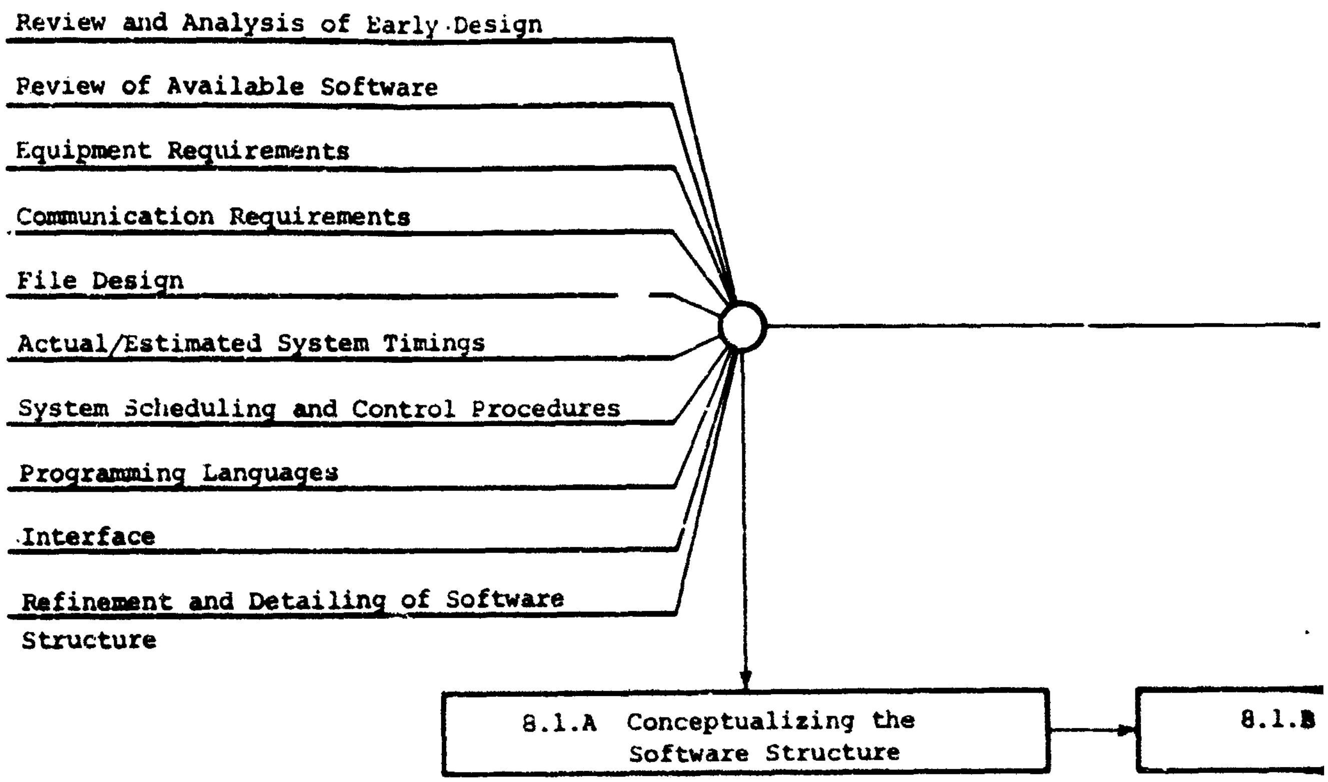
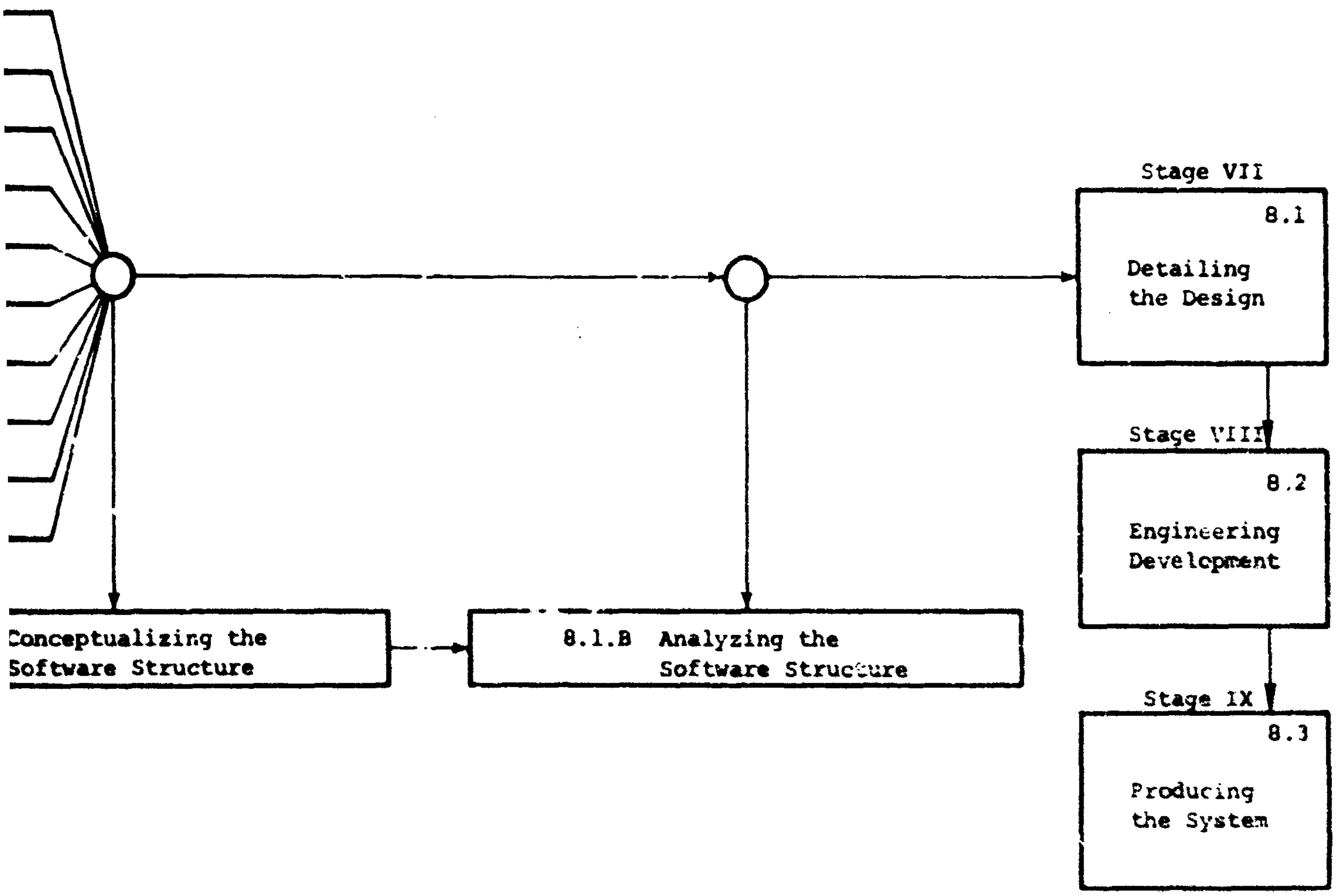


Figure 8-1. Overview of Software Design Engineering Procedures



eering Procedures

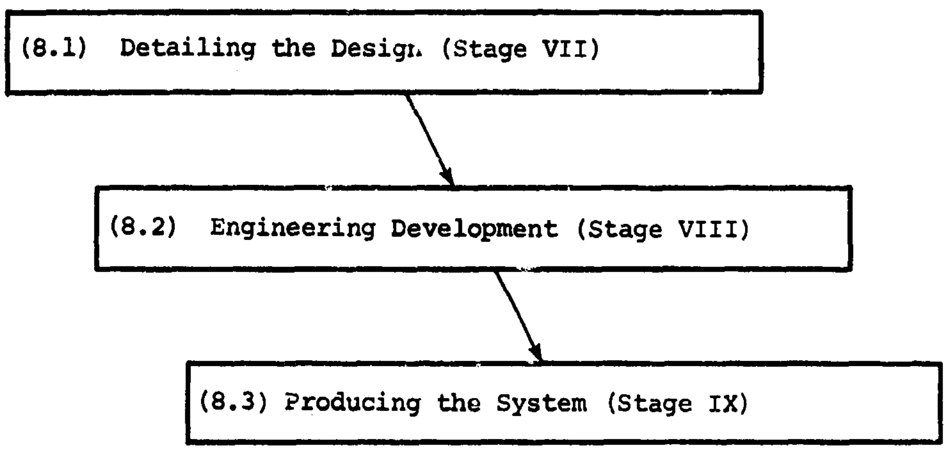


Figure 8-2. Stages of Design Engineering

1. Interprets operator commands and/or control cards which describe the application.

2. Gives control of computer to application programs in the proper sequence, handles job-to-job transition.

3. Schedules and performs input/output and related functions for application programs, relieving the programmer of hardware-oriented considerations and optimizing input/output device utilization.

4. Governs the operation of language translators (assemblers, compilers, linkage editors).

5. Provides program interference control.

6. Provides debugging and error diagnostic services.

7. Assigns physical input/output devices to logical files referred to by the programs: permits run-time substitution of alternate devices or accommodated changes in equipment configuration.

8. Handles program-to-program transitions that involve loading additional instructions into the computer from an external storage within a single job.

9. Provides dynamic allocation of core storage and other resources in a multi-programming environment.

10. Enforces the discipline required to run many programs at once in a time-sharing environment.

11. Allocates tasks to processes in a multi-processing environment.

The components of the operating system which perform the above functions are depicted in Figure 8-3.

A secondary product of the systems programming effort may be termed installation programs. These programs have the purpose of testing the operating system and the application programs. They are of three main types:

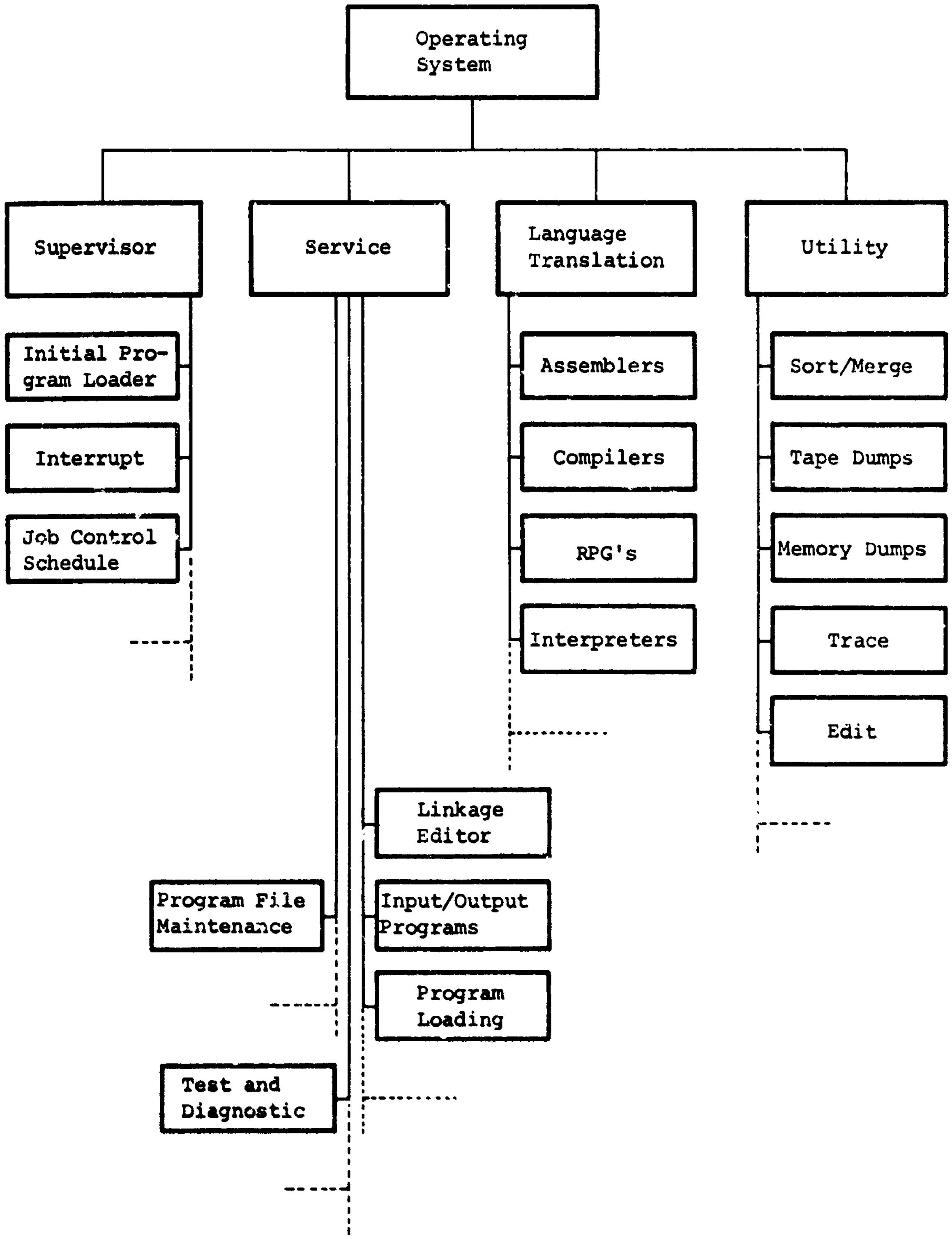


Figure 8-3. The Operating System

1. Operation system simulators.

2. Application system simulators.

3. Hardware simulators.

The remaining software elements are application programs. These programs are hardware plus operating system dependent. The operating system facilitates the applications programming effort, as well as providing control functions. The operating system also imposes limitations and restrictions upon application programmers since they must adhere to the operating system structure. Application programs, then, are those programs which perform the actual data processing tasks derived from system requirements and objectives.

8.1 Detailing the Design (Stage VII)

Just as the entire system development process is characterized by progressive iteration, that description is equally applicable to software subsystem development. Software development involves a complex exchange of information, analysis, assessment, and decision making. A simplified model of the major stages through which this complex of iterative activities eventually evolves is shown in Figure 8-4. The identified stages are described below.

8.1.A Conceptualizing the Software Structure

The approach to software conceptualization can be described in terms of some broad assumptions.

1. Software development is dependent upon at least tentative hardware configurations.

2. Selection of hardware is highly dependent upon software packages which:

    a. Accompany the hardware--that is, software which is indispensible to hardware operation (operating system in total or segments of the operating system).

(8.1.A) Conceptualizing the Software Structure

(8.1.B) Analyzing the Software Structure

Figure 8-4. (8.1) Detailing the Design (Stage VIII)

b. Are available in addition to the hardware-software package (additional operating system segment programs plus application program segments).

The emergence of the software conceptualization based upon the above assumptions is illustrated in Figure 8-5.

Software structure conceptualization is the process of describing how available technology is brought to bear upon the system functions allocated to the software subsystem. In order to structure the software, selected system elements must be focused upon in greater detail than in early design. The role of these system considerations is illustrated in Figure 8-6. Each major area of consideration is discussed below.

Review and Analysis of Early Design. Software-specific design is greatly facilitated if, at its inception, the implications of results from early design are clearly understood. Usually, an intensive review of concepts, data, and plans from early design is a productive initial step in structuring software. The aspects of early design worthy of careful consideration from the viewpoint of software development are functions allocation documentation and the design concept description.

It is inherent to the early design process that many tentative concepts and decisions are established on the basis of assumptions rather than on firm data. It is especially important to be aware of software-relevant assumptions of early design and of their implications for software. Although effective early design emphasizes an effort to explicate critical assumptions, there is almost certainly a multitude of implicit assumptions underlying early design. It is critical not only to consider the impact of assumptions explicitly stated in early design documentation, but also to determine what, if any, implicit assumptions involved in early design have a legacy for software design.

The software-oriented review of early design should have two main results:

1. To assure that software design can proceed within strictures and assumptions laid down by early design, to identify and explicate areas in which early design has resulted in assumptions or strictures which are incompatible with the imperatives

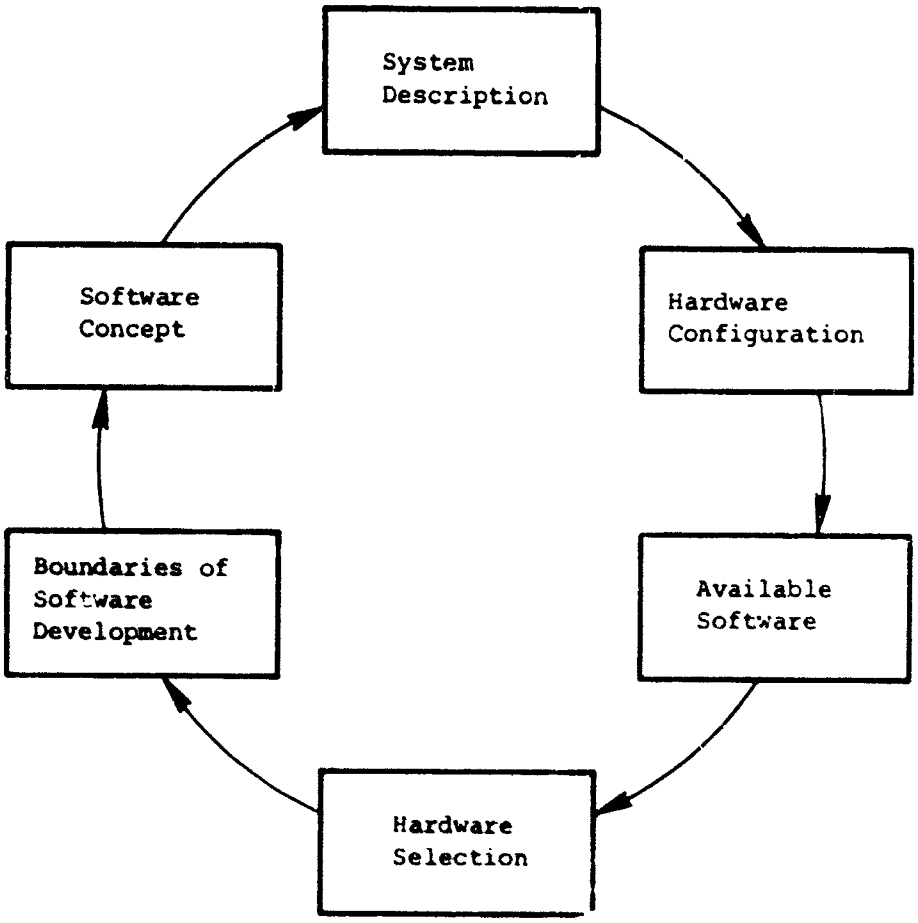


Figure 8-5. Emergence of Software Structure Conceptualization

of software design, or to identify areas of possible conflict between early design and the imperatives of software design so that definite resolutions can be established as priority requirements.

2. To assure that all of the productive efforts of early design relating to software are incorporated in the initial software concepts, designs, and plans.

Review of Available Software. The critical process involved in reviewing information concerning existing software beyond the extent of review in early design has a two-fold purpose. The first is to survey the possibilities of acquiring off-the-shelf software packages. The second is to identify models of successful solutions elsewhere which might help to guide in-house software development. These factors are considered concurrently with (1) the design and procurement of hardware, and (2) the software structure conceptualization.

Make-or-buy analysis. This process consists of matching hardware characteristics against software functions to determine the benefits of off-the-shelf software. A number of crucial considerations must be examined, including:

1. Immediate versus long-range needs.
2. Sophistication required for software maintenance under operational conditions.
3. Immediate performance capability versus future growth potential.

Make-or-buy analysis should at least exploit the information available from hardware manufacturers, user groups, and software companies. It involves consideration of cost and time of an in-house programming effort versus the cost of acquiring a package and performing necessary alterations. Relevant factors include the following:

1. In-house software development costs are frequently difficult to estimate.

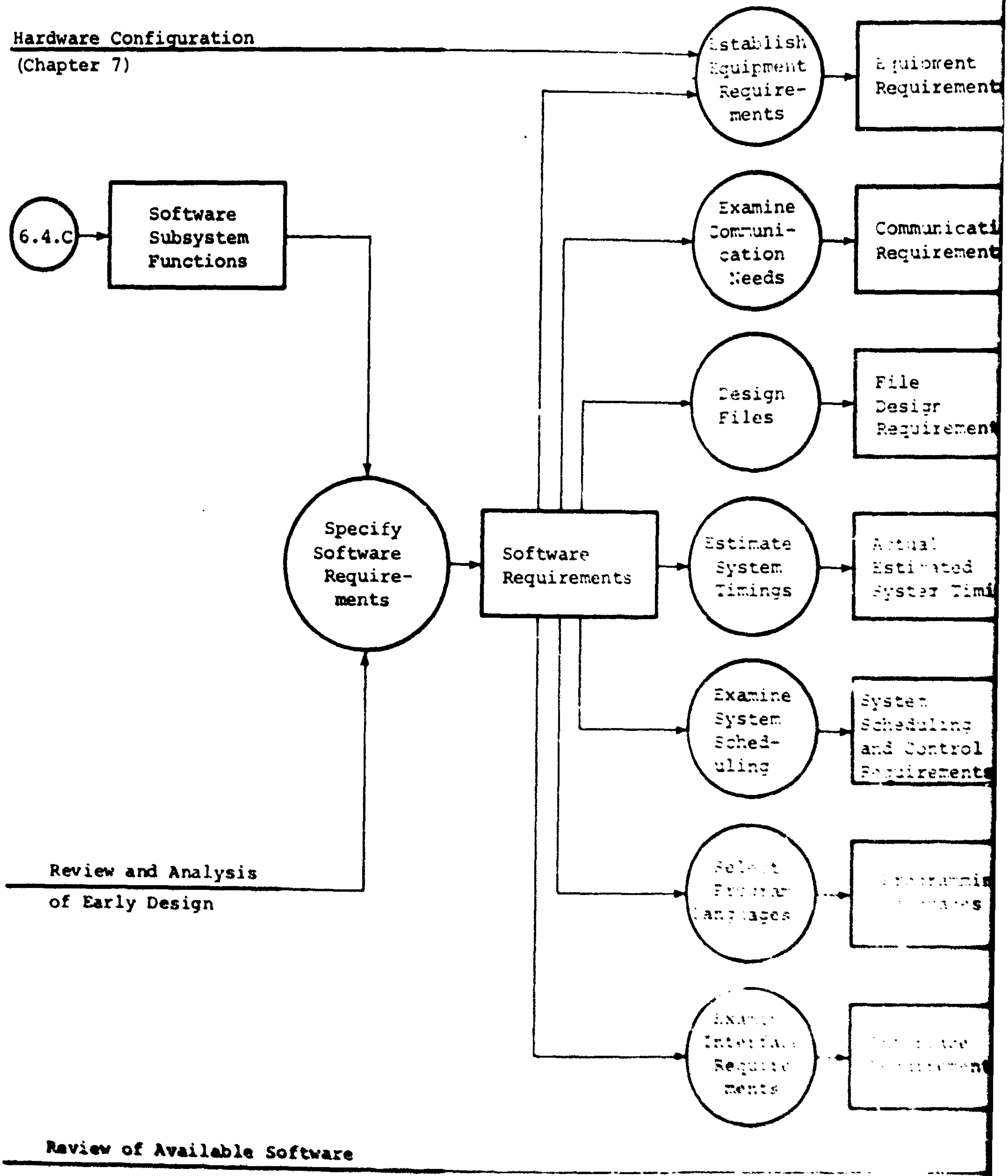
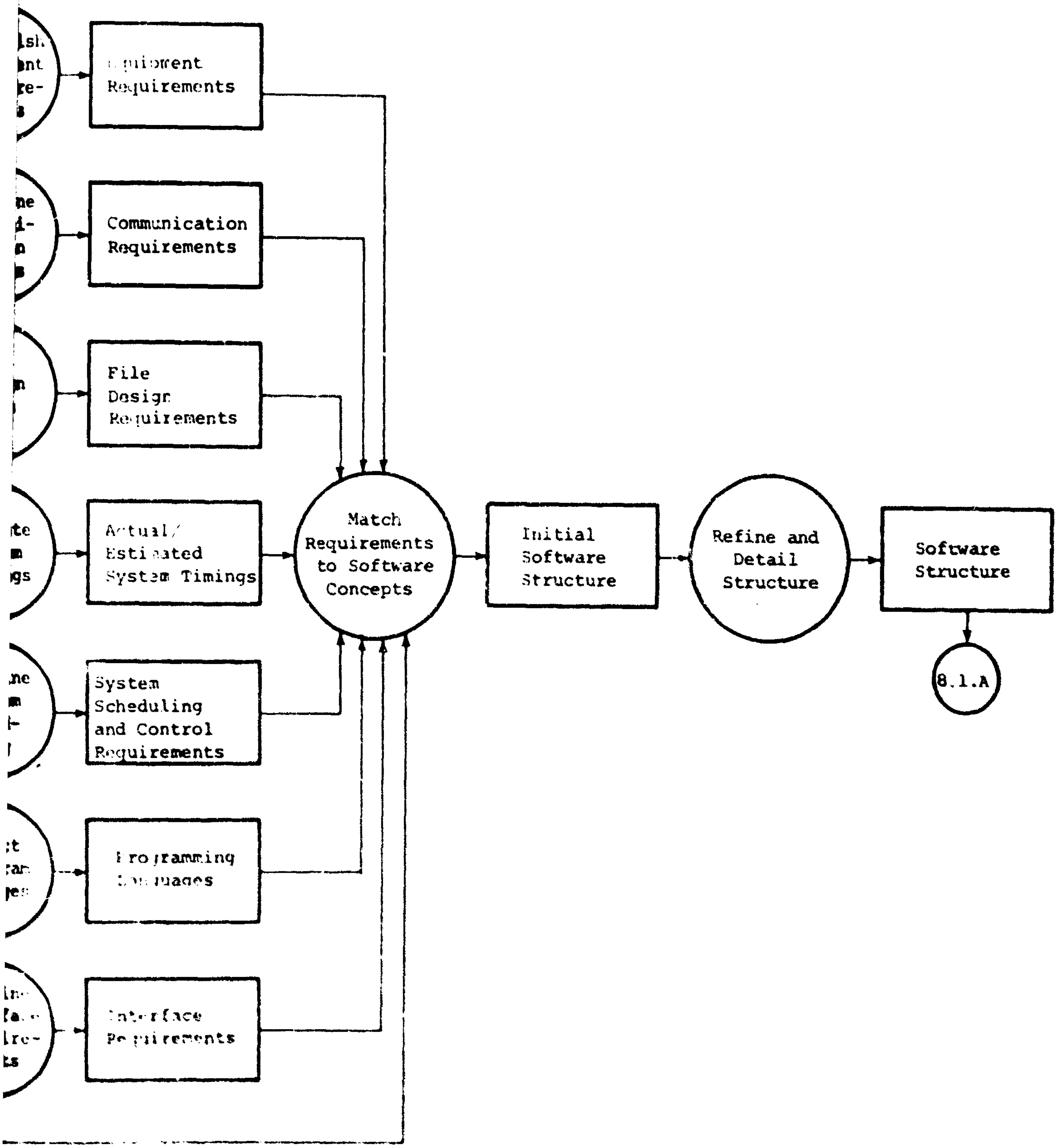


Figure 9-6. (8.1.A) Conceptualizing the Software



Requirements
Communication Requirements
File Design Requirements
Actual/ Estimated System Timings
System Scheduling and Control Requirements
Programming
Interface
Match Requirements to Software Concepts
Initial Software Structure
Refine and Detail Structure
Software Structure
8.1.A

2. Packages are immediately assessable in terms of meeting performance specifications--in-house software development may prove to be unsatisfactory upon completion.

3. Technical staff work load--higher priority system considerations may make software acquisition more desirable.

4. Programmers may be inclined to overestimate capabilities or be unrealistic in appraising available software in order to accept the design responsibility.

Specific criteria used to evaluate software packages include:

1. Package cost, including:

   a. Direct--initial cost (purchase price).

   b. Indirect--modifications, personnel, installation, operation, control and clerical procedures, conversion, production running of the package, and ongoing maintenance.

2. Quality, including a consideration of the developer's reputation, history of past delivery, and quality of earlier packages as well as the presumed capabilities of the package under current consideration.

3. Design, including:

   a. Necessary information files in master files.

   b. File organization and data management.

   c. Rerun and file protection.

   d. Control procedures, audit trails.

   e. Input/output options and requirements.

   f. Programming techniques.

   g. Flexibility of operation.

4. Generality and expandability, including:

   a. Sufficient additional capabilities or efficiencies.

   b. Possibility of additional features.

   c. Greater data volume processing.

   d. File growth.

These factors are weighed against user requirements. Package design tradeoffs to achieve generality and expandability can be implemented at the price of various inefficiencies such as throughput, file size, memory size, and equipment configuration.

5. Operational status, including:

   a. Extent of operational testing and installment.

   b. Throughput timings acquired from vendor--actual not theoretical.

6. Equipment configuration. The following factors are critical in matching software to hardware:

   a. Central processing requirements: accuracy, size, and speed.

   b. Sp[illegible]l features such as memory protection and decimal arithmetic.

   c. Input/output storage devices--type, speed, special features.

   d. Off-line equipment.

   e. Communication equipment.

7. Programming language, including:

   a. Matching language of package with existing software language and with programmer staff language capabilities.

b. Conversion between hardware--time and cost evaluation.

c. Matching application program language with operating system and operating system language with hardware.

8. Documentation. The extent of documentation reflects the quality of the package and should exist on four levels: systems, program, operations, and user.

9. Installation support, including:

a. Extent and quality of installation support--on-site assistance.

b. File conversion or creation.

c. Training--at all levels: clerical, operational, programming, systems, and management.

10. Ease of program maintenance.

Identification of models. The second purpose for reviewing existing software is to identify software configurations and development techniques in similar systems which have proven to be successful and efficient. Such configurations and techniques serve as models to help guide in-house programming efforts even when cost or other overriding considerations dictate that an existing package should not be purchased. There is, of course, a broad middle ground where portions of an existing package are procured and used directly, other portions are procured but augmented or modified, and yet other portions have to be generated essentially from scratch but along lines closely analogous to existing packages.

Equipment Requirements. The data processing equipment must be described, at the minimum, in terms of configuration requirements, including:

1. Central data processing components.

a. Size

b. Core time

c. Special features

2. Peripherals

a. Channels

b. Storage

c. Input/output devices

d. Speed

e. Buffers

3. Equipment interface--new equipment with old as well as with new.

The equipment configuration information is primarily a function of the hardware developers and the necessary documentation is derived from that source.

Communication Requirements. Depending upon the function allocations of early design and the extent of the communications system, software is responsible for at least the following communications aspects:

1. Initiation and control of data reception.
2. Assembling bits into characters and characters into messages.
3. Coding conversion (may or may not be the same as main computer).
4. Error checks.
5. Message editing.
6. Recognition of end-of-record, end-of-transmission characters.
7. Delivery of messages to main programs.
8. Acceptance of messages from application programs.
9. Preparation of messages for output.
10. Initiation of transmission.
11. Monitoring of the sending process.

12. Signaling end-of-transmission.

13. Fallback action.

14. Line control--interrupt mechanism.

The primary consideration in communications line functions is flexibility. In most systems, transmission requirements are modified either to meet changing needs or changing requirements. In general, then, control of communications input/output should be a programmed function rather than a hardwired function. The process involved here is to determine the method by which the software will perform its assigned functions. Review of previous systems and state-of-the-art is considered here.

File Design. For present purposes, a file is considered to be a collection of related records (adjacent data items, manipulated as a unit) which represent accumulated input available for processing.

File design is undoubtedly one of the most crucial aspects of system design. Two prime considerations are efficiency and reliability. The following aspects of file design are essential to creating an efficient and reliable file system:

1. Specification of processing requirements (application program system).

2. Specification of input/output requirements and characteristics derived from communication input/output definition and early design functions descriptions (format of each entry).

3. Classification of various data items.

4. Organization of data in storage.

5. Selection of storage media (cooperative hardware-software specification effort).

6. Distribution of data on storage media.

7. Use of directories or compools.

8. Mode of transactions (batched or unbatched)--derived from communication input/output definition and processing requirements.

9. Randomness of transaction processing.

Concurrent consideration is given to the following factors which determine, in addition to the estimated application needs, the file space demand:

1. Estimates of future growth.

2. Record compaction. A balance must be achieved between file size reduction at the expense of processing time and core storage. Various compaction techniques may be employed and are described in relevant literature.

3. Additions and deletions. Reorganization of the files may be necessitated periodically where the addition or deletion rate is great enough to require it. This rate is derived from analysis of system input. Where the addition or deletion rate requires additional file space, special attention must be given to the organization of the additions and deletions, the addressing scheme, and the method of file reorganization.

4. Addressing techniques. File addressing is the fundamental task of determining the organization and method of identifying the records in logical files. This is a problem of selecting the most cost/efficient method of file manipulation.

Relevant sources in file design are:

1. Lefkovitz, D. File structures for on-line systems. New York: Spartan Books, 1969.

2. Applied Data Research, Incorporated. A handbook on file structuring: Volume I. Griffiss Air Force Base, New York: Rome Air Development Center, 1969. (RADC-TR-69-313)

3. Applied Data Research, Incorporated. The representation of algorithms: Volume II. Griffiss Air Force Base, New York: Rome Air Development Center, 1969. (RADC-TR-69-313)

4. International Business Machines Corporation. <u>File design handbook: Final report</u>. San Jose: San Jose Research Laboratory. Gaithersburg, Maryland: Federal Systems Division, 1969.

5. International Business Machines Corporation. <u>File organization modelling system: User's manual</u>. San Jose: San Jose Research Laboratory. Gaithersburg, Maryland: Federal Systems Division, 1969.

<u>Actual/Estimated System Timings</u>. Closely related to file design efforts are timing and synchronization studies which determine system capabilities in handling transactions or messages captured from the signal environment and processed by the system. In general, the message queues which must be examined are of four types:

1. <u>Input queues</u>--messages which arrive at the system and which are queued for processing.
2. <u>Channel queues</u>--requests for input/output operations or system channels.
3. <u>Process queues</u>--items on which processing has begun and interrupted and which await processing completion.
4. <u>Output queues</u>--output messages produced by the system which are queued for transmission.

Queue estimation must attend to the following relevant factors:

1. <u>Limiting the number of items in the queue</u>. If queues are maintained in core storage, the amount of core and block chaining mechanism should be examined.
2. <u>Location of queue</u>. The queues described above can all be located in core. Other possibilities include non-real-time queues and overflow queues on disk, tape, and drum as well as input/output queue capacity in communication devices.
3. <u>Length of items in queue</u>. The analysis involves determining whether fixed-length or variable-length items are essential.

A tradeoff is generally made between control (in fixed-length queues) and flexibility (in variable-length queues).

4. Queue priority structure. The queue mechanism is evaluated in the context of the message environment.

5. Size of queue. Consideration of this aspect of system queues is necessary for basic decision making concerning the programming of queues. It may be necessary to utilize simulation techniques used to analyze the queues under various system conditions. A special simulation language may be required. An alternative method is queuing theory. A reference for queuing analysis is:

   ARINC Research Corporation. Guidebook for systems analysis/cost-effectiveness. Annapolis: Author, 1969. (AD 688 154)

System Scheduling and Control Procedures. The prime consideration here is an examination of the system operating modes and control procedures. The operating mode refers to the conditions or methods of operation of a device. Determination and description of the operating mode is essential to the conceptualization of software both in the non-structured software component of early design and in the software subsystem. The operating mode describes the functioning of the system in the environment and its utilization of time.

If the system operates in a complex signal environment, interrupt mechanisms and multi-programming become primary considerations. Interrupt refers here to the temporary termination of current processing activity due to a hardware event of some sort which necessitates operation in another mode. Typical interrupt conditions are:

1. Initiation of input/output operation.
2. Termination of input/output or file operation.
3. Error or malfunction signal.
4. Onset of special condition.

The automation of interrupts depends upon hardware sophistication. Software development can be required to perform all interrupt events as outlined below:

1. Inhibit further interrupts until completion of priority routine.
2. Locate the termination point in application program to be stored for return.
3. Preserve unit contents (register, accumulator) needed by application program.
4. Determine cause of interrupt as well as the mechanism of interrupt.
5. Determine required action and transfer of control to program which will accomplish requisite action or queuing of action request.
6. Transfer control to point of interruption in the application program.

Multi-programming here refers to the operation of application programs on different transactions and with varying degrees of interdepencence in various stages of simultaneous execution. The necessary information for software development concerns:

1. The number of possible or required transactions which can be processed.
2. The relationship of input/output timings to process timings.

In the generally complex operating environment of on-line information systems in particular, systems and application software is an essential design aspect. The on-line information system environment combines the following characteristics:

1. The system serves a large number of users in multiple locations.
2. User interaction with the system is limited in comparison to input/output terminal or communication capacities.

3. The system performs a variety of functions, including at the minimum:

   a. Large file data storage and retrieval.

   b. Computations.

   c. Message processing and forwarding.

4. The on-line system is in a continual state of improvement and expansion.

5. On-line system reliability requirements are stringent.

6. The system produces a continual demand for additional operational hours.

These environmental characteristics provide a number of relevant software design implications with respect to the operating mode of on-line information systems.

1. Software can be utilized to demultiplex incoming user data whenever possible.

2. Software must be designed to handle user input data with highest priority in regard to file modification.

3. On-line programs can provide sophisticated and complex user identification schemes when required.

4. Software design is particularly important in off-line systems designed to duplicate on-line system operations because of the following:

   a. Off-line software maintains a duplicate of on-line files.

   b. Off-line software is used for assembling and checking out new programs.

   c. Off-line software provides essential services which keep the system operational such as compiling statistical data on service characteristics; editing, sorting, and merging new user data; checking system files, etc.

d. Off-line software can simulate user input traffic and check system output.

e. Off-line and on-line software, if exactly similar in organization and content while operating in different modes, provide maximum system reliability.

5. On-line software must assist in error detection, recovery, and switch-over timings.

Programming Languages. Program languages may vary across a broad spectrum from heavy orientation to the machine to an orientation heavily toward the user's natural language. Software for the operating system, for applications, and for query (on-line, interactive dialog) probably differ substantially in orientations--as illustrated in Figure 8-7.

The software language problem is essentially this:

1. The software language must be solidly anchored in machine language, for the machine is often rigidly demanding of the form in which it accepts final instructions. Particularly for the operating system, the less assembly and compilation (translation) required, the more efficient are operations.

2. The region of tradeoff on the query or interactive end is likely to be relatively narrow. User acceptance of specialized language requirements resulting from the machine's needs is probably low. Rightly, the user often demands that the software accommodate his natural operational language when he interacts with his data base. The personnel and training demands are generally excessive if operations personnel are required to use languages having any significant machine orientation.

3. The interesting and broad region of tradeoff is with respect to language for applications programming. The possible range is from heavily machine to natural language orientation. At the machine-oriented end, minimum investment in assembly and compilation is required--but the

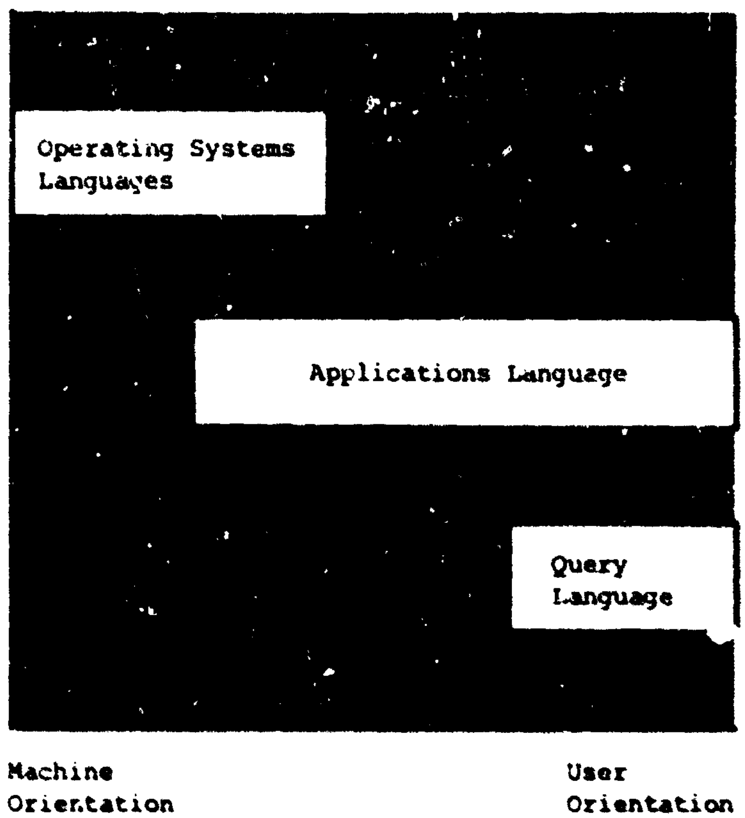


Figure 8-7. Schematic Representation of Relationship Between Software Purpose and Natural Language Characteristics

specialized skills demanded of the programming staff are maximal. At the natural language end, minimum specialized programming skills are required, but the assembly and compiling demands to translate programs into efficient machine instructions are maximal. Particularly if user personnel are to play a key role in development of application programs, it is imperative that the initial programming effort be accomplished in language oriented heavily toward the natural language of the user.

4. The general trend in computer system design is toward much more facile and inexpensive automatic translation of user-oriented languages into efficient machine instructions. Almost with each passing day, the arguments for programming (other than the hard core of the operating system supplied by the vendor) in machine-oriented language diminish.

5. There are strong reasons to keep the number of programming languages (particularly at the point of initial program development) to a minimum.

6. The impact on skill requirements for program maintenance must be taken into account as well as requirements for programming personnel in development.

7. All of these different, and sometimes conflicting, considerations must be balanced or reconciled in some reasonable fashion.

Two principal sources for language evaluation and selection are:

1. Wegner, P. Programming languages, information structures and machine organization. New York: McGraw-Hill, 1968.

2. Rosen, S. (Ed.) Programming systems and languages. New York: McGraw-Hill, 1967.

Interface. The software interface encompasses three general areas: software-personnel, software-hardware, and software-software. The primary concern here is with the last of these since the software-hardware interface

has been described in a previous section and the software-personnel interface is more appropriately dealt with in Personnel. Both, however, are of concern to the software structure conceptualization.

The software-software interface refers to the interface between systems programs and application programs and the connection between existing systems and projected systems. The systems application program relationship must be specified in detail. The application program effort generally proceeds somewhat independently of systems programming since operating systems can usually be purchased off-the-shelf in total or in segments. Therefore, the operating system must be well defined since application programs continually utilize and refer to the operating system. Generally, the linkage between the operating system and application programs, as well as between programs and subroutines, is accomplished with macro-instructions. The macro-instructions are either included in the software package or may be an in-house programming effort. Regardless of their source, macro-instructions should be a primary consideration in software development as they perform such functions as obtaining core areas for application programs and releasing the area upon program termination, executing input/output operations, and transfer of control to the operating system.

Refinement and Detailing of Software Structure. The conceptualization of software structure, then, is tied to early design efforts and later detailed examination of the above areas. With increasing specification and analysis, it is possible to formalize an overall software structure conceptualization. This conceptualization is based upon the:

1. System requirements specified in the above areas.

2. Available software review.

3. Hardware configuration.

The software structure conceptualization will have the following essential characteristics:

1. State the systems programming requirements of the system in enough detail to enable careful evaluation of operating system software packages and to specify simulation needs.

Definition areas should include the:

a. Operating System

1) Supervisor Programs
2) Service Programs
3) Language Translations
4) Utility Programs

b. Simulation Programs

1) Operating System Simulation
2) Application Simulation
3) Hardware Simulation

2. Identify the application programs as logically independent parts of the system derived from the system data processing requirements. The level of detail should permit evaluation of application packages.

3. Identify program interface requirements.

8.1.B Analyzing the Software Structure

This critical aspect of software development occurs concurrently with the conceptualization effort. The object of the conceptualization analysis is to evaluate the proposed system software in terms of cost/effectiveness. The process is essentially a feasibility study and is summarized in Figure 8-8.

The software structure analysis, modifications, and subsequent "best fit" structure selection involves the following considerations:

1. Determination of resources available including time available, machine, and personnel capability.

2. Examination of current system and future systems.

3. Review of other systems in operation or available.

4. Identification of data manipulation capacity of concept and evaluation of the capacity with reference to system requirements and objectives.

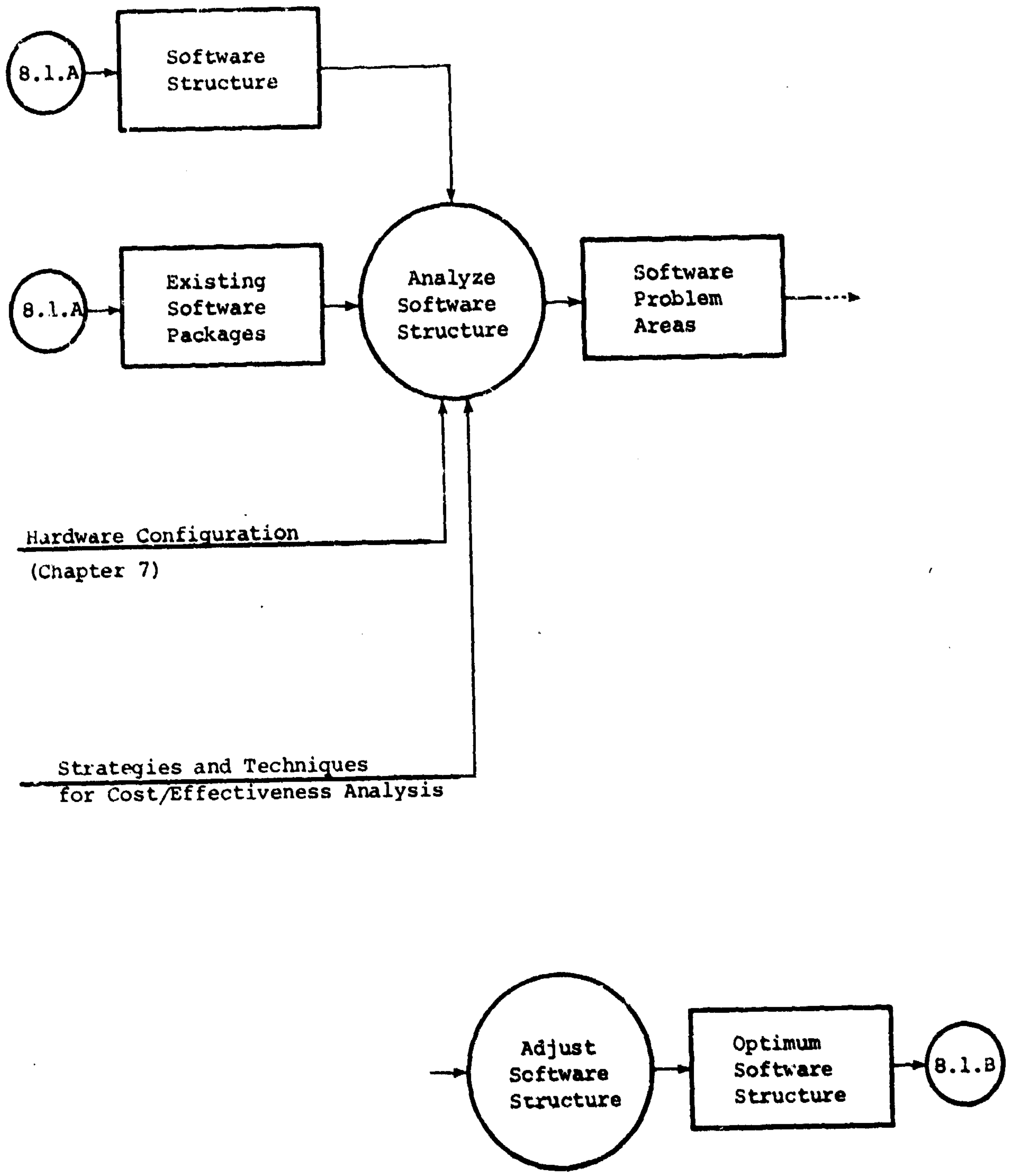


Figure 8-8. (8.1.B) Analyzing the Software Structure

The analysis process is referenced to the general characteristics of the system which determine cost/effectiveness. The methods of Design Assessment are largely applicable here. The result of the iterations of analysis and conceptualization produces an optimization, or at least sufficiency, of the gross software structure design.

8.2 Engineering Development (Stage VIII)

Engineering the development of software elements involves the reduction of the structure to a symbolic and/or physical representation sufficient to demonstrate that it is workable in an operation-like environment. It is obviously impractical to make this demonstration with a full system. Software pieces are rationally integrated to estimate probable characteristics. Simulations of the operational conditions are often less than full fidelity. Engineering development of software must also be coordinated with parallel efforts in the hardware and personnel subsystems to ensure that all potential interface problems are exposed.

Engineering software development results in fully detailed software specifications which permit purchase or production of programs. Figure 8-9 depicts the progression of activities which results in these specifications. The following factors are considered in arriving at final software design specifications:

1. Implications for total design.
2. Initial cost.
3. Efficiency of the structure.
4. System performance requirements.
5. Operating costs.
6. Software availability.
7. Personnel requirements--operational as well as programming.

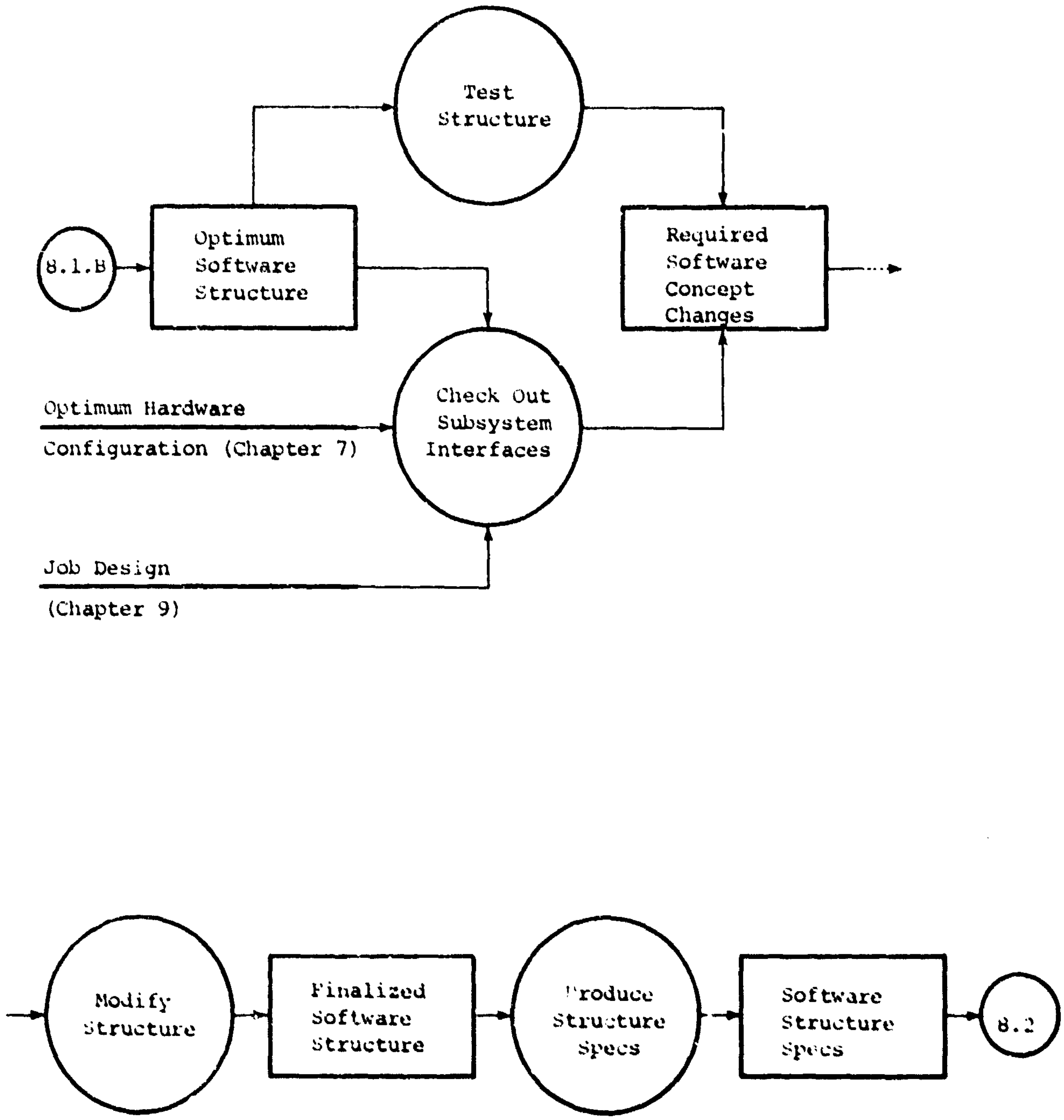


Figure 8-9. (8.2) Engineering Development (Stage)

8.3 Producing the System (Stage IX)

In this phase of software development, the system hardware is largely taken as a "given." System and application software requirements are matched against available vendor software. Then, the segmentation of software development separates the total programming effort into "chunks" in reference to personnel, cost, and time factors. The segmentation may be based on the following considerations:

1. Grouping of related or dependent functions.
2. Isolating independent functions.
3. Grouping processes according to file activity.
4. Grouping processes according to document production.
5. Phasing when related or dependent processing procedures are grouped.
6. Phasing where an independent processing procedure is isolated.
7. Simple division according to memory size.
8. Simple division according to estimated relative run time.
9. Inquiry speed.

Segmentation of software development may be necessitated by:

1. Adoption of existing application programs.
2. Use of general-purpose software packages.
3. Programmer capacity.
4. Testing requirements.
5. Installation requirements.
6. Maintenance and modification requirements.

The program development/procurement process embodies generalizable phases of activity which result in software operational status. These stages consist of translation, design, production, and integration, as illustrated in Figure 8-10. The focus here is upon the translation and design phases from which the

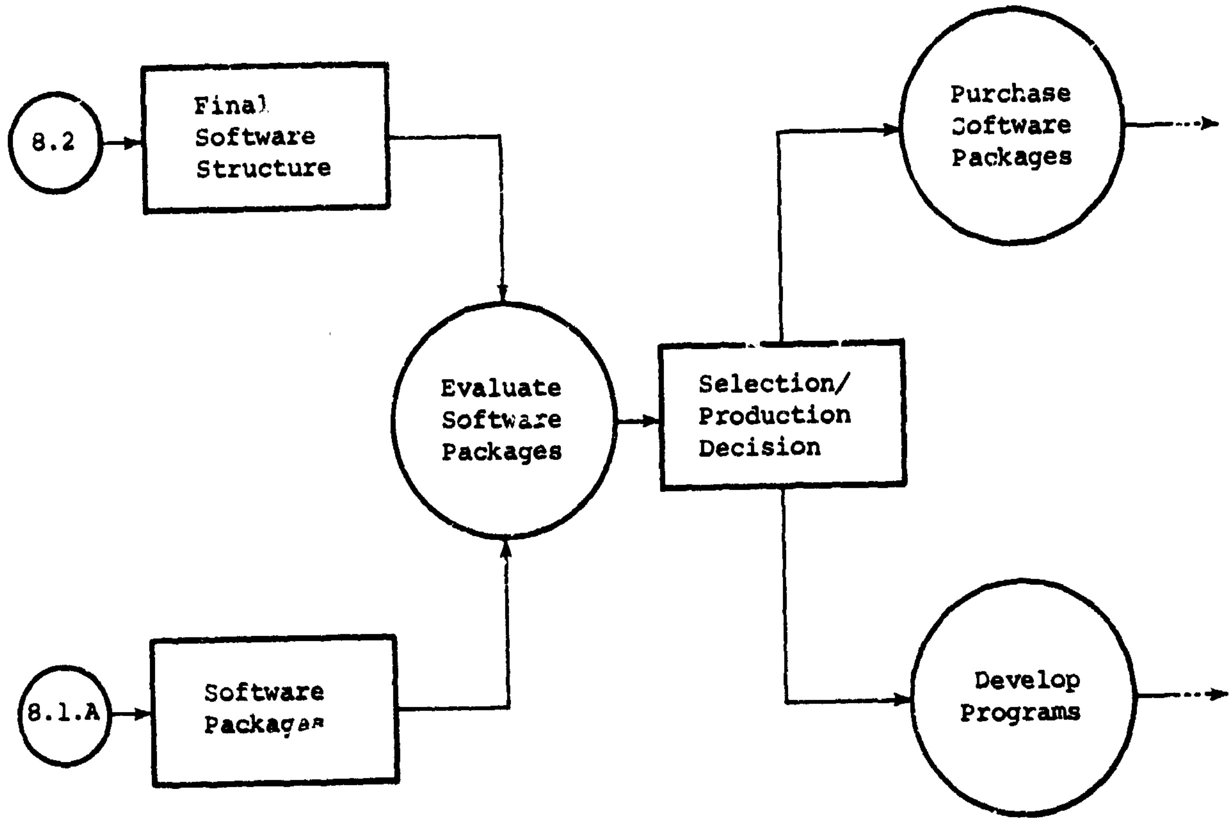


Figure 8-10. (8.3) Producing the System (Stage IX)

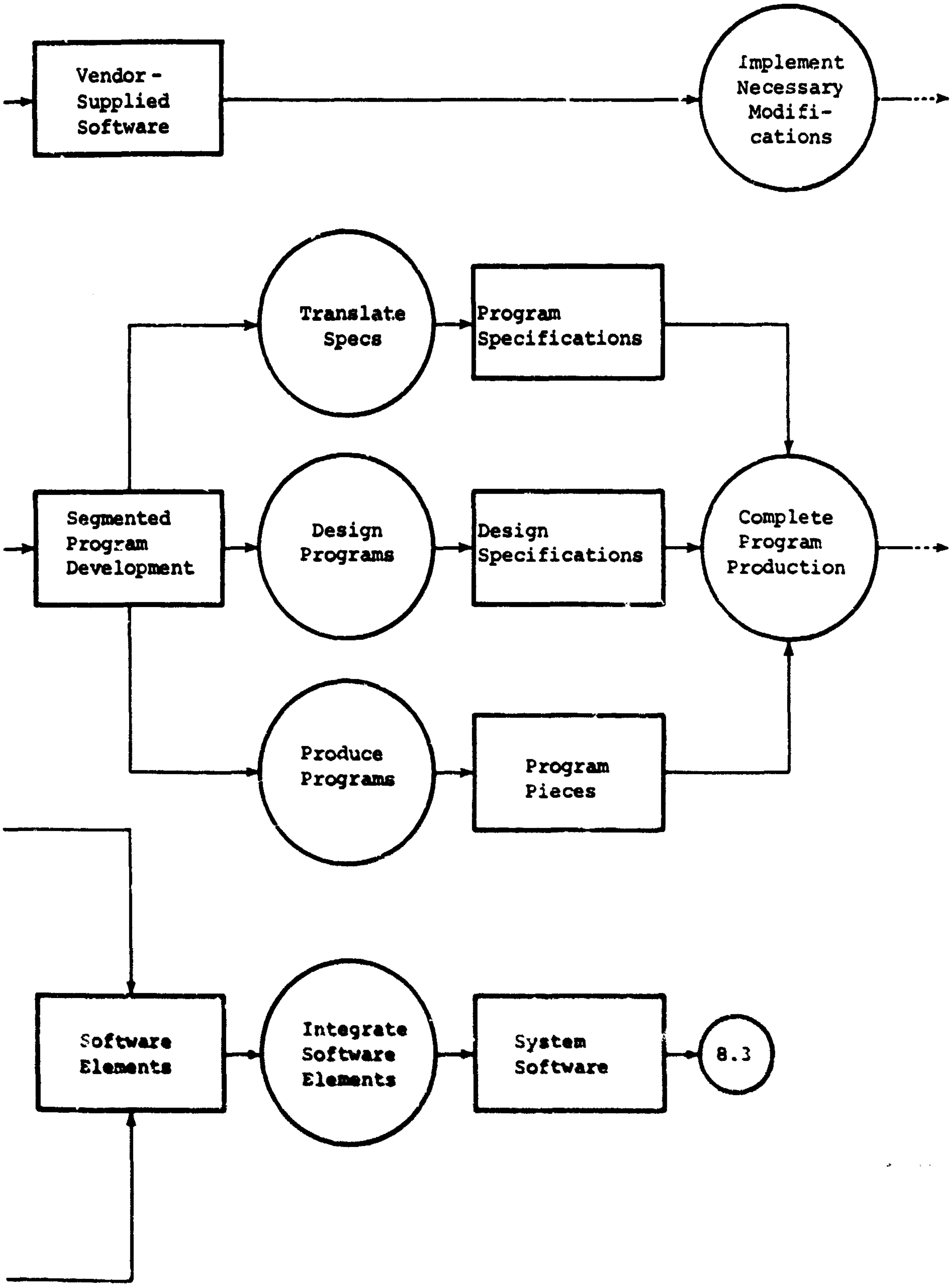


Figure 8-10. (Continued)

technical program writing effort proceeds. Production refers to the flow charting, coding, and individual program checkout activity. Installation of the program system requires a performance assessment of the compatibility of systems programs with application programs (program integration). The installation programs (simulators) serve as the means for this evaluation and are usually vendor/user supplied. The total systems checkout is beyond the province of the software development subsystem and requires a reunion of the three subsystems into an integrated installation and operational shakedown effort.

The program development effort is directed in either of the two functional program areas--systems or applications. The systems programming effort is initiated with a management make-or-buy decision (based upon software technical advice) which reveals the extent of vendor-supplied software. In-house systems programming is required to:

1. Alter system program features to meet particular system needs.

2. Develop programs to meet deficiencies in vendor-supplied operating system and installation programs.

Generally, the gap in systems software will fall in the utility and service program segments of the operating system.

The applications programming effort is the primary focus of program development in that the application programs meet system performance objectives. Again, application programs may be vendor supplied but, in general, are system specific. Thus, the major programming effort is in the applications area. Regardless of functional area, program development is characterized by the following phases.

Translation. This phase is concerned with describing the software structure requirements in programming terminology. It translates gross conceptual flow charts of the software structure into detailed program specifications. The program specifications should:

1. Identify and resolve inconsistencies in structural conceptualization.

2. Organize the structure requirements in logical groups.

3. Detail the inputs, processing requirements, and outputs for the programs.

4. Describe the nature of interface in system.

5. Define any necessary special features (error detection and correction backup procedures, limitations, and restrictions).

The primary purpose of the translation phase is to detail the system software requirements in a programming orientation. The initial program specifications should continually be referenced to the overall software structure conceptualization and the hardware-software package configuration.

Design. Program design is the most significant phase of the programming development effort. Design specification expands the program specification in technical detail and comprehensiveness. The focus of the design phase is to produce an integrated program system which has incorporated all foreseen and planned system contingencies. The design stage has the following essential characteristics:

1. Provides detail such hat program logical statements can be translated one for one into the specified programming language.

2. Interfaces between manual and automated equipment are described in detail.

3. Describes individual program input and output.

4. Details file requirements and specifications.

5. Specifies timing and core storage restrictions.

The design phase results in complete identification of the relationship of individual programs to system objectives and requirements. That is, the system processes can be described in terms of a corresponding program or subprogram. The design specification completes the specification of the relationship of the software subsystem to the operating environment. The product of the design phase is a logical flow chart or decision tables of the specified program such that:

1. All major decision points are identified.

2. Decision criteria are identified.

Production. This phase of program development which consists of program coding, checkout, and documentation activities is considered to be of a technical nature well described and examined in numerous sources. It is sufficient, then, to note that program production is a necessary prerequisite for assessment of software compatibility.

Integration. The compatibility of all software subsystem elements in systems and applications areas is, of course, a primary consideration throughout the software structure conceptualization and eventual program development phase. The operational compatibility must be actually evaluated in a simulated system environment prior to total system installation and shakedown. Three general simulation areas are necessary:

1. Operation system simulators.

2. Application system simulators.

3. Hardware system simulators.

Hardware and operating system simulation requirements are generally the province of the hardware vendor. The hardware simulations must represent the internal computer, input/output, and peripheral device configuration. The operating system simulation should be capable of evaluating system software performance under controlled traffic conditions in terms of accuracy, reliability, and response time.

Application program simulation is specified concurrently with application programs and must have the following essential characteristics:

1. Measure the ability of the programs to perform data processing functions.

2. Measure system run time.

3. Interact with various operating system segments.

4. Represent error-handling capacity.

When the compatibility of the software elements is assessed and required adjustments and modifications incorporated, the subsystem effort merges to install and test the total system. The software capability is then evaluated along with hardware and personnel factors in the operational environment.

CHAPTER 9

DESIGN ENGINEERING - PERSONNEL

This chapter deals with the nature and impact of personnel considerations in system design. To build a balanced and integrated system, you must be able to accurately assess the extent of human capabilities in the system environment once functions are allocated to men, machines, and personnel. This chapter is aimed at identifying the factors which must be considered to ensure that requisite human capabilities are available to support system operations, and that human limitations are accounted for in the design of the system. Those factors are examined within the three design and development stages that comprise Design Engineering--detailing the design (Stage VII), engineering development (Stage VIII), and producing the system (Stage IX). An overview of the personnel subsystem development is presented in Figure 9-1.

The content of this chapter describes the development of the personnel subsystem, the activities involved and the information required. You will not emerge as a human factors specialist upon using this material, but you will have a broad perspective of the factors and activities necessary to build a personnel subsystem. It is important, then, for you to know what the personnel subsystem includes. Often used in a narrow sense, the personnel subsystem may refer only to system operators and the necessary training, selection, and proficiency measurement that accompanies the integration of such personnel into the system environment. A broader definition is used here. For our purposes, the personnel subsystem includes all human factors areas in system design and development. The term human factors denotes the interactive man-machine-environment relationships of the system with emphasis upon the human aspects of the interactions. Human factors refers to the application of methods, behavioral principles, and principles from related scientific fields to the development and evaluation of man-machine systems. Certainly human factors considerations enter into early design, but not until design engineering do the considerations become defined and established as a subsystem. Essentially, the personnel subsystem encompasses three main areas of concern:

1. Identification and Analysis of Human Performance Requirements--the process of translating functions allocated to

men into structured performance requirements. These performance or task requirements must be identified, described, and analyzed from the functions allocations. The tasks are then clustered into positions, jobs, and occupations (job design), and the manpower needs of the system are projected concurrently with job design. Some of the factors involved in determining human performance requirements are:

a. The nature of the tasks necessary to perform system functions.

b. Behavioral considerations associated with the tasks.

c. Effect of new system tasks upon existing tasks and their impact upon the present administrative organization.

d. Efficient combination of tasks into position, job, and occupation groupings.

e. Spatial and temporal arrangement of the task groupings.

f. Manpower requirements derived from tasks and job design efforts.

2. Human Engineering--the determination of facts about human behavior, the development of systematic methods for considering human characteristics in the design of system elements, and the application of these facts and methods throughout system design efforts. Human engineering involves the optimization of equipment and facility design, manual and job-aid design, and form and code design.

3. Personnel Selection, Training, and Proficiency Measurement--the identification of human capabilities necessary to meet system performance requirements, the development of training methods and techniques determined to be essential for

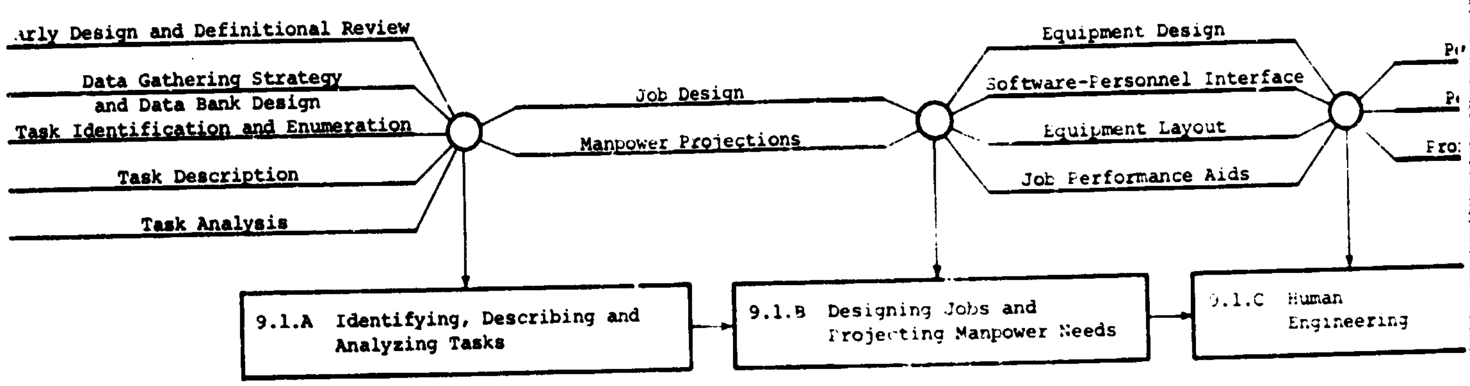
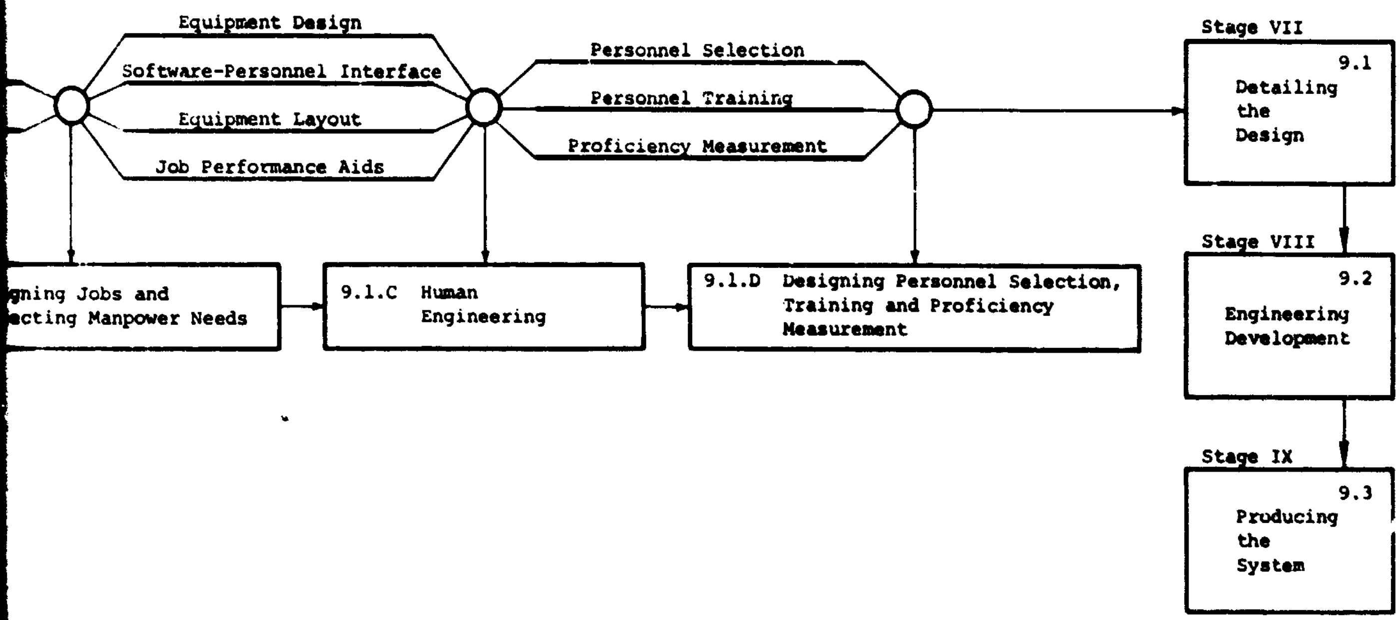


Figure 9-1. Overview of Personnel [illegible]


Equipment Design
Software-Personnel Interface
Equipment Layout
Job Performance Aids
Personnel Selection
Personnel Training
Proficiency Measurement
Stage VII
9.1
Detailing the Design
Stage VIII
9.2
Engineering Development
Stage IX
9.3
Producing the System
gning Jobs and
ecting Manpower Needs
9.1.C Human Engineering
9.1.D Designing Personnel Selection, Training and Proficiency Measurement

Procedures

task performance, and the creation of evaluation criteria for measuring performance. The major emphasis in this human factors area is to develop a personnel organization which is compatible with the automated elements of the system and to ensure that human resources are efficiently utilized in the system environment. This requires adequate development and description of the following areas:

a. Requisite personnel skills, knowledge, and experience derived from system tasks.

b. Performance criteria and design of performance measures.

c. Selection standards and methods.

d. Specification of training objectives.

e. Formulation of training strategies and development of materials.

f. Sequencing and evaluation of training.

In dealing with these human factors areas, this chapter stresses the interdisciplinary nature of personnel subsystem activities, organizes personnel considerations in a structured context, and conceptualizes the flow of personnel subsystem development activities. This model of the emergence and growth of the personnel subsystem should be of greater usefulness than examination of a sample of the almost unlimited human variables which influence system operations. Therefore, no attempt is made to detail human factors analyses, describe human factors experimentation, or analyze specific behavioral parameters such as perception, identification, and interpretation which characterize human functioning in information systems. Instead, it is assumed that you will support this outline of personnel subsystem development with technical methodologies and analyses described in available resources. It is also assumed that you will make adjustments for the particulars of your system in utilizing the chapter information. The focus here, then is upon the relationship rather than the specifics of personnel subsystem components and the integration of the personnel subsystem with hardware and software development.

The broad levels of personnel subsystem development activities are illustrated in Figure 9-2. The nature and implications of the development activities and the essential characteristics of each are treated in the remainder of the chapter.

9.1 Detailing the Design (Stage VII)

Detailing the design concept produced by early design efforts involves working from a general concept to a level of detail where a needed personnel function capability is known already to exist or can be specified. The major activities involved in detailing the personnel subsystem design are presented in Figure 9-3.

9.1.A Identifying, Describing, and Analyzing Tasks.

The initial focus of the personnel subsystem effort is upon developing detailed human performance requirements, collectively termed task requirements. Task requirements are derived from the personnel functions allocated in early design and evolve in greater detail and specificity as development efforts progress. As elemental building blocks for the personnel subsystem, task requirements are a preliminary or at least concurrent consideration to job design and manpower forecasting, human engineering, and personnel selection, training, and proficiency measurement.

Deriving task requirements involves several interrelated activities which result in a detailed description of human performance in the system. The nature of each activity is described below and illustrated in Figure 9-4.

Early Design and Definitional Review. Although the personnel subsystem breaks out from hardware and software only after the completion of early design, some personnel considerations are important prior to early design completion in defining system resources and constraints and in allocating functions. Functions allocation requires considerable human factors information to determine, at least tentatively, the capabilities and efficiencies of man versus machine and program in performing system functions. The reason for a review of early design analyses and definition is to examine the allocation decisions

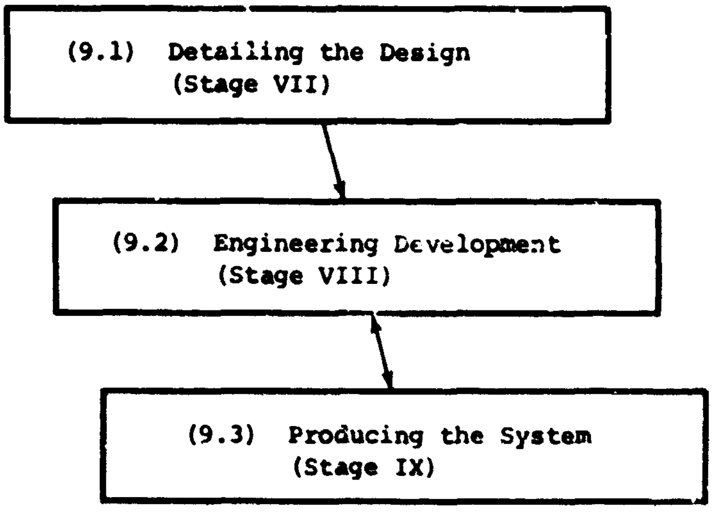


Figure 9-2. Stages of Design Engineering

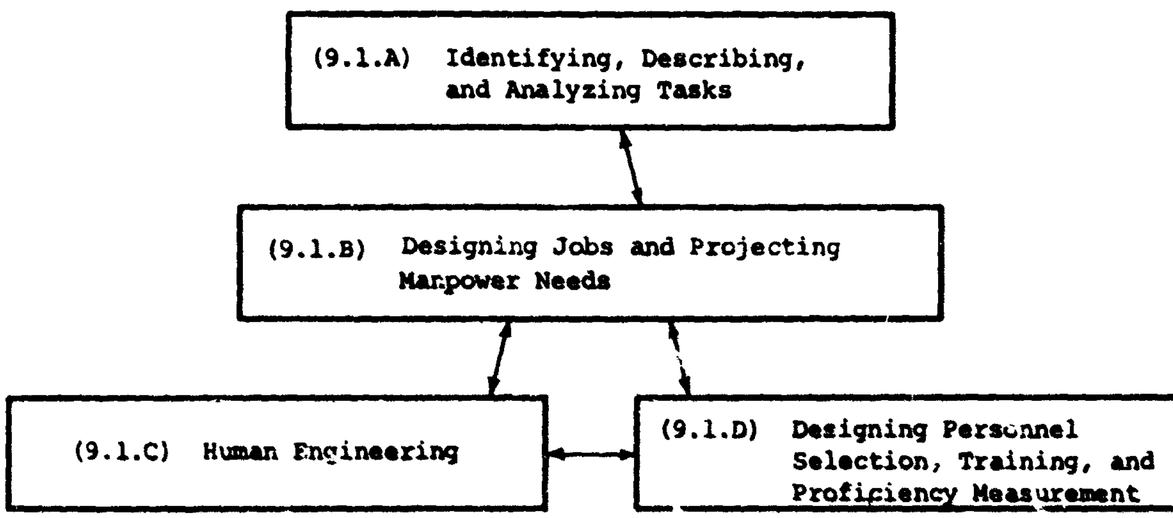


Figure 9-3. [illegible]ailing the Design (Stage VII)

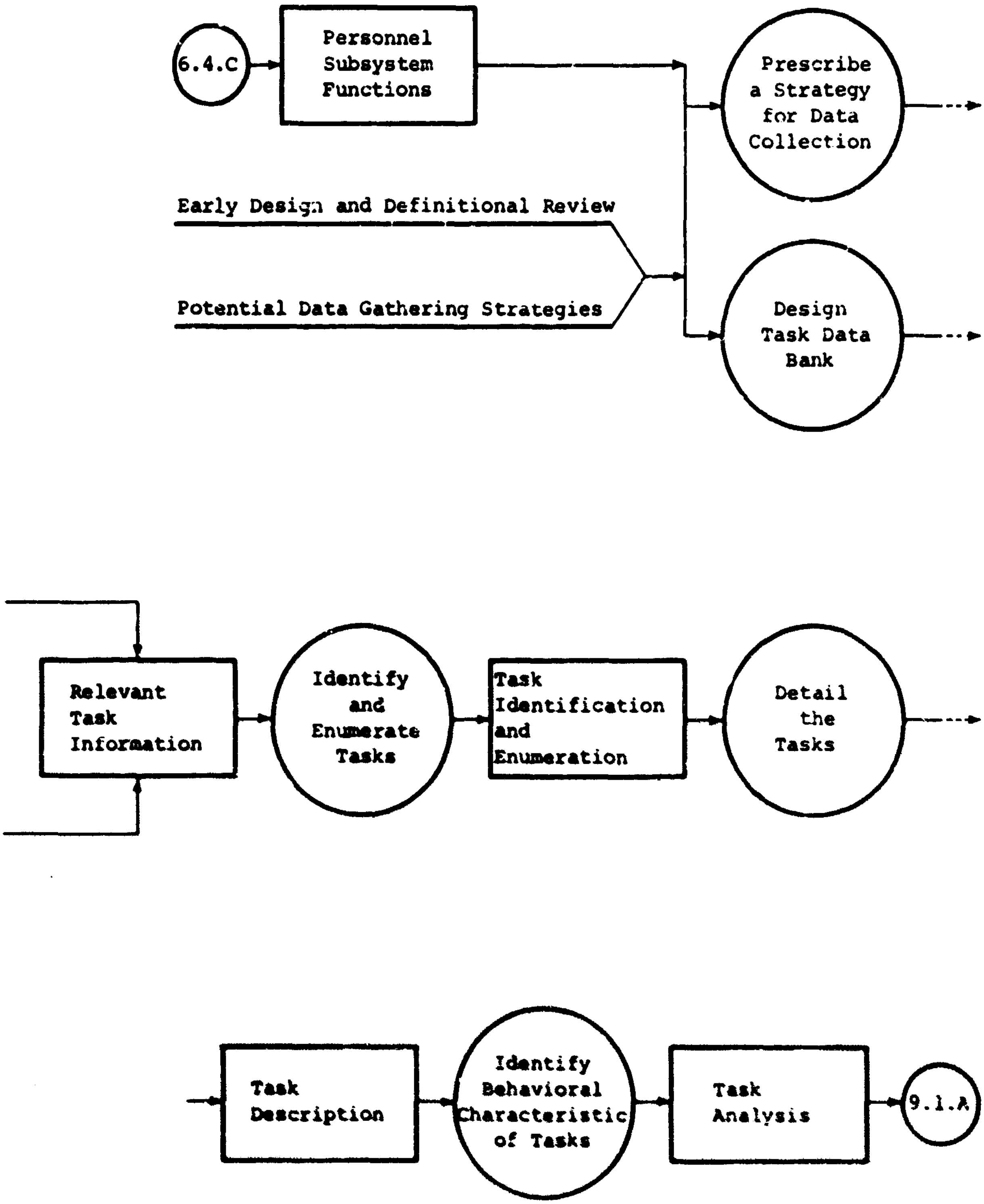


Figure 9-4. (9.1.A) Identifying, Describing, and Analyzing Tasks

to ensure that human capabilities match allocated functions. Hardware-personnel tradeoffs, as well as software-personnel tradeoffs, are relevant design considerations at the onset of the personnel subsystem effort. The essential areas of information which may be used as evaluation criteria in this regard are described in the functions allocation stage (6.4) of Early Design.

Here are two important factors in assessing the desirability or necessity of man-machine-program tradeoffs:

1. Establish equipment and program reliability and maintainability in relation to the availability and cost of the personnel necessary for its operation and maintenance.
2. Determine that the investment and operating cost of automation are appropriately optimized against manpower availability and cost.

The review of early design activities encompasses an examination of the nature of personnel-oriented considerations and assessment of the implications of personnel functions in regard to system objectives and requirements. The design concept documentation is the primary source of reference for such a review.

Data Gathering Strategy and Data Bank Design. It would be both difficult and inefficient to derive task requirements for the system without first developing a strategy for collecting the task data and designing a means for organizing the data. The necessity for accurate and comprehensive data is apparent since the derivation of task requirements precipitates the entire personnel subsystem development efforts. Therefore, a major concern must be to develop an adequate and efficient data collection strategy. Some possible strategy ideas can be found in Data Methods. The selected strategy depends mainly on the availability of the data and the time allocated for collection. To ensure that the data, once gathered, are easily accessible, consideration should be given at the same time to the design of a task data bank. The data bank design depends largely on the nature of the data, their volume, and the available storage media.

So once the preliminary organization steps have been taken, the information gathering can begin. The most reasonable sources of task data are listed in the next segment of the chapter. Since the task data sources include software and hardware configurations, the data gathering strategy should be developed for utilizing available information from all parallel design and development efforts.

<u>Task Identification and Enumeration</u>. The process of task identification and enumeration consists of identifying discrete groups of behaviors directed toward a specifiable outcome required to meet system objectives. The emphasis is upon relating personnel functions to the units of equipment, programs, and operations with which the functions are to be performed. An essential requirement of task identification is that the task must have a definite system output. The task, then, can eventually be described in terms of stimulus inputs, decisions, and response outputs.

Some potential sources of task identification and task enumeration data are as follows:

1. System objectives and requirements.
2. System performance requirements.
3. Mission analysis.
4. Functions analysis and allocations.
5. Equipment configuration conceptualization.
6. Software structure conceptualization.
7. Interface descriptions.

<u>Task Description</u>. Task descriptions follow the identification of tasks within the system and specify the precise nature of the interactions of man with machine and with the system environment. A task description states what must be done by system personnel if a given function or subfunction is to be accomplished. The essential purpose of task description is to relate functional requirements to personnel requirements. Task descriptions are generally used to detail the nature of the personnel and system environment relationship in regard to:

1. Software.

2. Hardware.

3. Facility layouts.

4. Other personnel.

Task descriptions are utilized as a basic reference for the remainder of the personnel subsystem design and serve as immediate input to task analysis activities. Since the task descriptions specify human performance requirements, performance criteria used to evaluate system personnel are derived from these data. In order to function in the capacities just described, the task description should contain the following information, specified in detail:

1. Immediate purpose of task.

2. Specific equipment output.

3. Human inputs.

4. Decisions involved.

5. Required outputs to accomplish the stated purpose.

In addition, the process of task description generally includes the following activities which are necessary to accurately and comprehensively describe the nature of the personnel tasks within the system:

1. Match descriptive formats and tasks.

2. Designate task identifiers and titles.

3. Identify within-task action sequences and alternatives.

4. Describe individual action components.

In general, task descriptions should specify along an operational time scale the cues that the individual should perceive and the related responses which he should make.

<u>Task Analysis</u>. While task descriptions specify the tasks to be performed by the system personnel, task analysis results in a model of system performance in terms of behavioral elements. Task analysis is the systematic

study of the human behavior parameters or characteristics necessary to accomplish the task. A fundamental purpose of task analysis is to furnish design criteria for input to human engineering efforts concerning the nature of hardware-personnel interface in controls and displays and equipment layout. An additional function of task analysis is to provide information regarding the personnel selection and training requirements resulting from the specification of behavioral and psychological aspects of the task. Thus, task analysis is usually a basic source of data for the personnel subsystem areas listed below:

1. Job design.
2. Preliminary manpower estimates.
3. Subsystem interfaces.
4. Equipment and facility design.
5. Selection criteria.
6. Training and training aids.
7. Performance measurements for training and system evaluation.

Task analysis generally includes the following activities:

1. Estimating the criticality of the task for system performance.
2. Estimating likely errors and probable performance levels in the task.
3. Identifying emergency contingencies.
4. Describing relationships to other tasks.
5. Estimating skill, knowledge, and experience requirements.

The resulting data from task analysis may reveal that a task derived from functions allocations to the personnel subsystem may be beyond human capabilities or available personnel resources. In such a case, alternative

methods for meeting the task requirements must be determined. Upon completion of any necessary allocation tradeoffs, the personnel subsystem development effort breaks out into three interdependent but distinguishable human factors areas: job design and manpower projections, human engineering, and personnel selection, training, and proficiency measurement.

9.1.B Designing Jobs and Projecting Manpower Needs

The personnel subsystem development effort may be characterized at this point by the following factors:

1. Design activities have resulted in the finalized allocation of system functions to personnel.
2. Identification and description of the tasks which must be performed have been accomplished.
3. Task analysis has specified the human characteristics tied to performing the tasks.

The objectives of job design and manpower projection which depend on the above design activities are to organize tasks into positions, jobs, or occupations and to specify the personnel requirements necessary in the resulting organizational structure. Job design is the allocation of tasks to position, job, or occupation units. Manpower projection is the identification of personnel numbers, skills, knowledge, and experience required by the positions, jobs, and occupations. The interdependent nature of job design and manpower estimation results in a more or less simultaneous achievement of two related objectives and is illustrated as such in Figure 9-5. For present purposes, the characteristics of each activity will be discussed separately.

Job Design. The differentiation of tasks into positions, jobs, and occupations is based upon the following conceptual distinctions:

1. Position--one or more tasks which must, for practical purposes, be performed by a single individual at a given location within the system.
2. Job--all of the tasks performed by a given individual at one or more locations within the system.

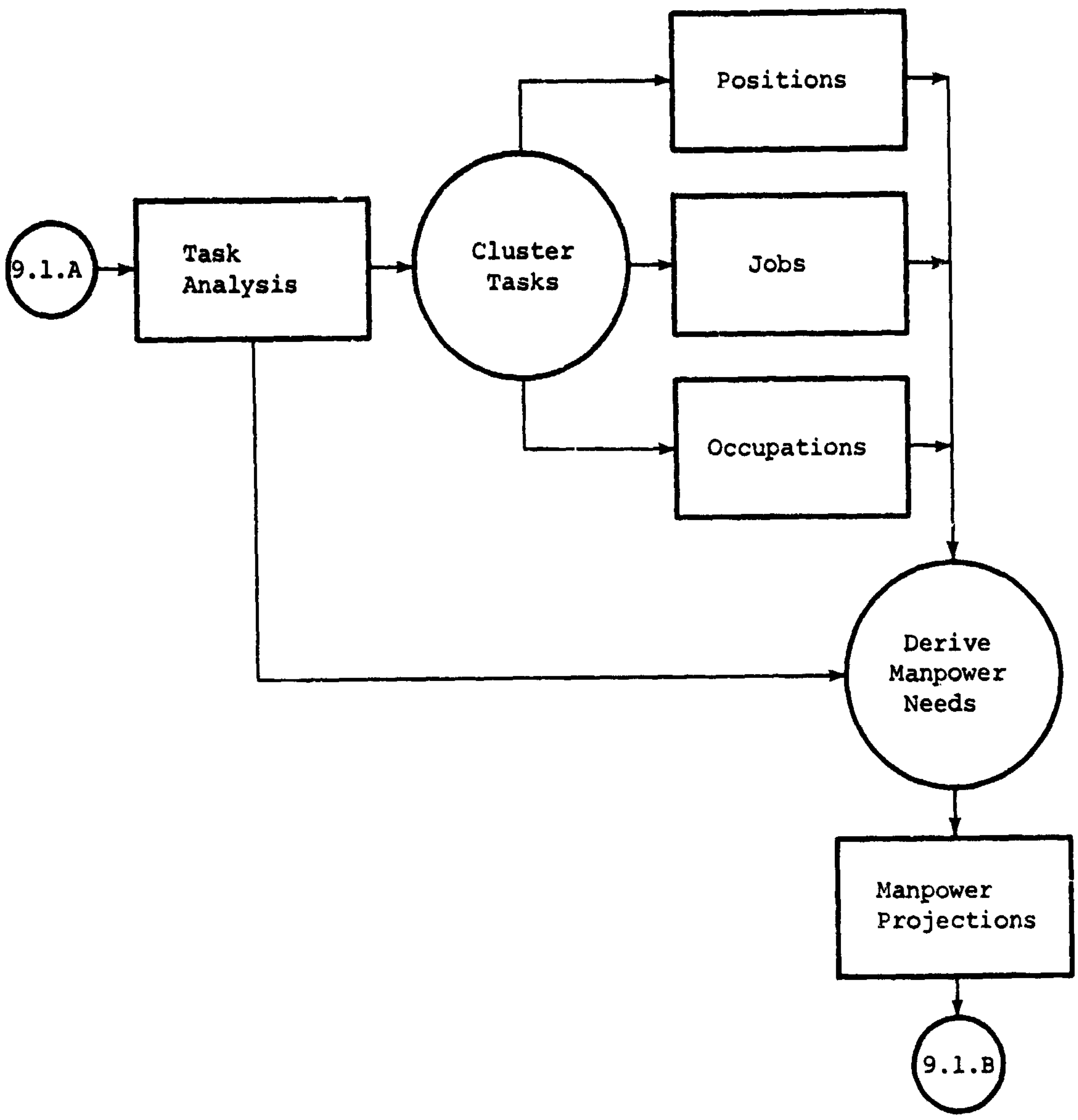


Figure 9-5. (9.1.B) Designing Jobs and Projecting Manpower Needs

3. Occupation--a family of related jobs involving a high degree of overlap in the tasks performed.

Task clustering or grouping is the central concept in job design. The criteria employed to cluster tasks into positions, jobs, and occupations are:

1. Common functions or objectives.
2. Location of performance.
3. Timing sequence of performance.
4. Equipment utilization.
5. Communications requirements.
6. Common capability requirements.
7. Level of difficulty and performance requirements.
8. Nature of the task in terms of its:
   a. Frequency.
   b. Criticality.
   c. Priority.
9. Time requirements.

The design of position, job, and occupation units aims to:

1. Maximize utilization of existing personnel classifications and job organizations.
2. Minimize training requirements.
3. Minimize additional or complex skills, knowledge, and proficiency requirements.

Parallel developmental efforts in the hardware and software subsystems are also important considerations in efficient job design. The hardware configuration and software structure have human engineering implications for job design. Task groupings must be referenced to the performance requirements of the tasks in terms of the physical location, equipment to be operated, the use of displays and controls, the program demands, and the communication media which form the operational setting of the task. Job design must be integrated

with equipment characteristics and computer program operations to ensure that the total interaction is compatible with human capabilities and that hardware and software conceptualizations have been evaluated as possible sources of constraint in job design.

Manpower Projections. With the emergence of an organizational structure of system tasks assigned to personnel, it becomes possible to derive estimates of the personnel characteristics and qualifications required for various organizational units. Manpower projections fulfill three essential purposes:

1. To define the specific numbers, qualifications, and locations of personnel required to implement the particular system being developed or analyzed. In this context, manpower projections are a specified developmental product to be used for operational planning.

2. To provide input to the selection, training, and evaluation requirements of the personnel subsystem.

3. To provide staffing guides or criteria such that system management will have a normative basis for supporting their judgments.

The manning projection effort may be characterized by a shift in design emphasis from the qualitative aspects of the personnel of the system to an integration of qualitative and quantitative considerations. The manpower estimates become increasingly refined as additional quantitative information becomes available. The numbers as well as skills and knowledge of personnel are specified.

Manpower projections generally involve the following activities and information about the system operational environment:

1. Comprehensive description of and provision for the activities performed by system personnel.

2. Identification of system contingencies with evaluation of manpower required to perform emergency actions.

3. Assessment of the operational schedule of the system in terms of sufficient personnel necessary to maintain operational status.

4. Identification of the number of different jobs in the system (job design efforts should attempt to minimize the number of different jobs).

5. Determination of the minimum number of personnel to meet system operational requirements.

6. Assessments of the minimum training requirements necessary to meet performance specifications.

Manpower projections guide developmental efforts in the human factors area of personnel selection, training, and proficiency measurement.

9.1.C Human Engineering

Comprehensive personnel subsystem development must consider human behavior principles in the interface of man with the equipment, software, facility, and operational environment of the system. Human engineering involves analyzing human behavior and applying human behavior characteristics throughout personnel as well as software and hardware efforts. Before human engineering can be initiated, however, the nature and extent of personnel performance requirements must be known. That is, the identification, description and analysis of personnel tasks is required to determine what personnel will do in the system. Human engineering can then be utilized in the design of equipment for operability and maintainability, computer program and personnel interface, the design of the physical environment, and the design of job performance aids.

The relationship of human engineering to those system components is discussed below and is illustrated in Figure 9-6.

Equipment Design. The human engineering emphasis in equipment design generally occurs in vendor coordinated design efforts. The personnel subsystem is largely concerned, then, in providing human engineering data which permits human factors evaluation of the hardware configuration prior to final hardware selection. In the case that hardware components are designed and developed

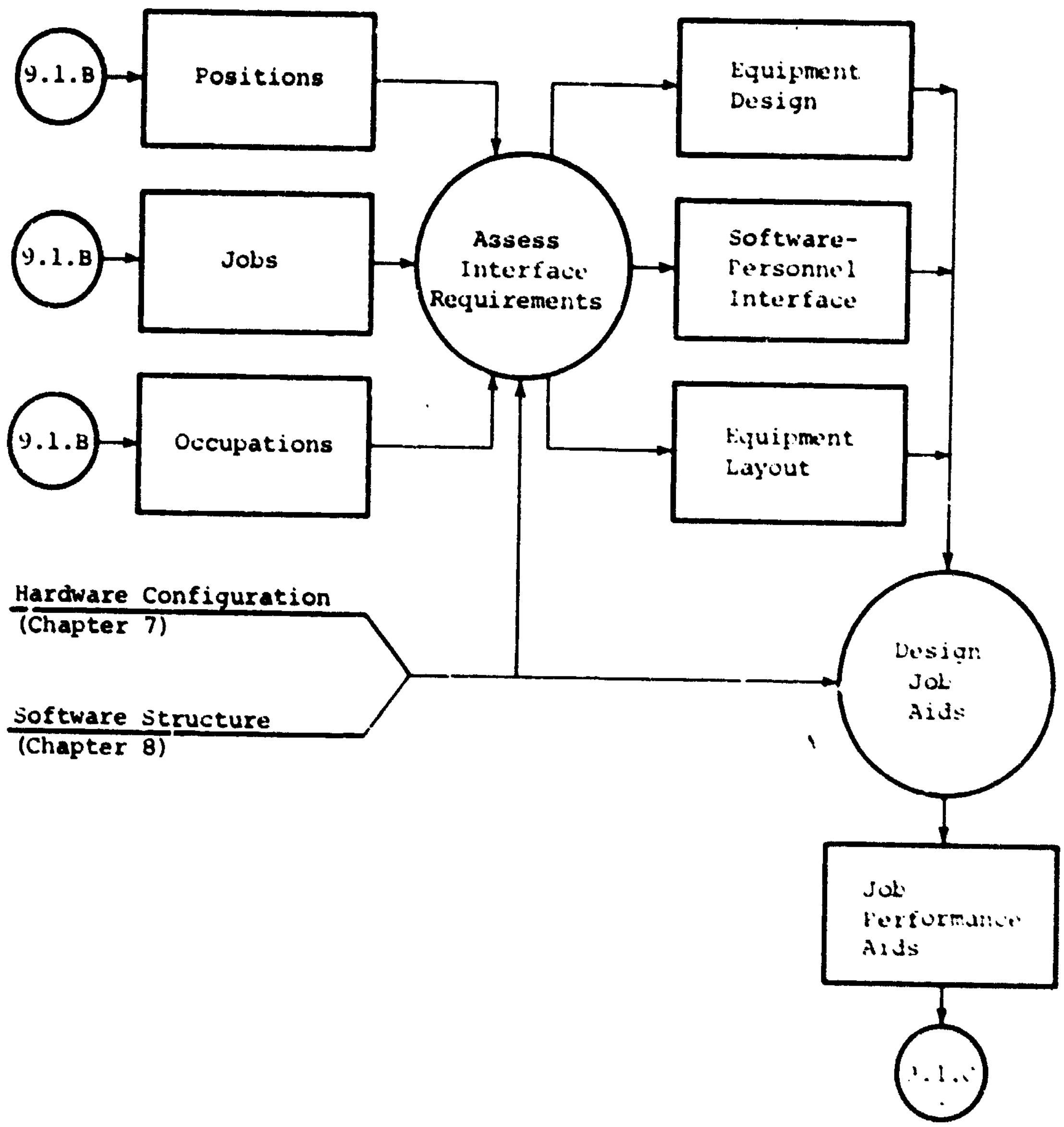


Figure 9-6. (9.1.C) Human Engineering

specifically for the system environment, then human engineering considerations contribute significantly to the design specifications. Both conditions require adequate human engineering technology. Listed below are specific features of equipment which generally require human engineering in information systems.

1. Display design.

   a. Extent of necessary sensory discrimination.

   b. Mode of sensory stimulation employed.

   c. Organization of combination display data.

   d. Form coding design.

   e. Censoring information inputs.

   f. False signal elimination.

2. Control design.

   a. Accessibility.

   b. Functional arrangement.

   c. Differentiation.

      Frequency
      Criticality
      Skill requirements
      Logical groupings
      Operational goal groupings

   d. Safety.

   e. Operational status.

   f. Spatial proximity of controls and displays.

3. Maintenance design.

   a. Displays and controls.

   b. External accessibility.

   c. Test points and equipment.

   d. Internal accessibility.

   e. Facility design.

   f. Personnel requirements.

Software-Personnel Interface. In general, human engineering of software design and development should attempt to achieve these goals:

1. Standardization--input media and output formats should be standardized for groups of computer programs designed to perform a specific function.

2. Automation--where computer programs perform system operations more rapidly and efficiently than human beings, error will be minimized and procedures simplified.

3. Decision control--the control of non-critical operational decisions can be efficiently and effectively handled by program control to minimize error possibilities in the data processing flow.

Equipment Layout. The design of the physical environment of the system utilizes human engineering to optimize the efficiency of man's performance. The physical environment design should minimize error potential in the system while incorporating safety and comfort features. So that equipment design or configuration does not entirely dictate the facility design, the physical environment must receive concurrent development attention.

The general sequence of activities and areas of principal concern in achieving an optimum physical environment design are outlined below:

1. Identify special user requirements in the system.

2. Identify external conditions.

   a. Equipment configuration.

   b. Natural and man-made environmental characteristics.

3. Identify relevant physical parameters and tolerances.

4. Evaluate equipment and environment design in terms of identified physical parameters and tolerances.

5. Supervise environmental development activities.

6. Assess design of physical environment.

Job Performance Aids. A job performance aid fulfills a unique, significant purpose with respect to the human component of man-machine systems: It acts to support or maintain man's performance within the limits established for overall system performance. Job aids include any device or technique, such as a light-pen, manual, or checklist which augments man's capabilities to perform the required tasks. Thus, an aid may eliminate, simplify, clarify, or implement decisions and actions required of system personnel in much the same way that a stored program enhances the performance of a central processor. In this respect, job aids generally complement training and can substitute for special training particularly where maintaining performance within strict limits is critical. For all these reasons, proper design of a job aid depends on the specific task characteristics, performance requirements, and operational conditions concerned. Furthermore, a balance should be achieved between job aids and training in order to produce the most effective personnel performance. To a large degree, this balance results from cost/effectiveness tradeoffs between the information content supplied by training and the optimal design of a job aid.

Job aid design should consider:

1. The application of human engineering principles to assure cost/effective results.
2. The overall information needs of the personnel subsystem--that is, for aids such as complete, accurate system operating and maintenance manuals.
3. The recurrent weaknesses in conventional aids (which must be offset or eliminated) such as: Unnecessary complexity; failure to delineate contingency conditions; incomplete and/or late; and awkward to use under actual conditions.

9.1.3 Designing Personnel Selection, Training, and Proficiency Measurement

The major objective of personnel selection and training efforts is to achieve and maintain a level of performance from personnel which satisfies system requirements. Proficiency measures must be developed to assess the

success of both section and training activities in meeting this objective. Figure 9-7 depicts the main activities involved in designing this subsystem area.

While selection and training procedures are complimentary and related personnel considerations, it is useful to distinguish the two in the following manner:

1. Selection introduces personnel into the system who either have:

    a. Requisite skills or knowledge, or

    b. Aptitude for required skills and knowledges.

2. Training procedures are applied to selected personnel in order to:

    a. Create requisite skills and knowledge, and

    b. Optimize human capabilities.

The distinction between these two personnel subsystem areas is rather arbitrary and serves primarily for discussion purposes. Indeed, selection and training operate in an interdependent manner in the system. The nature of their interdependence is apparent in the formulation of selection and training needs, both commonly derived from job design information. That is, the design of selection and training procedures, as well as proficiency measures, utilizes the same information sources which fall in four general areas:

1. Physical requirements of position, jobs, and occupations.

2. Informational requirements of position, jobs, and occupations.

    a. Broad knowledge.

    b. Specific knowledge.

3. Job skills in terms of output and performance requirements.

4. Personality characteristics required for performance.

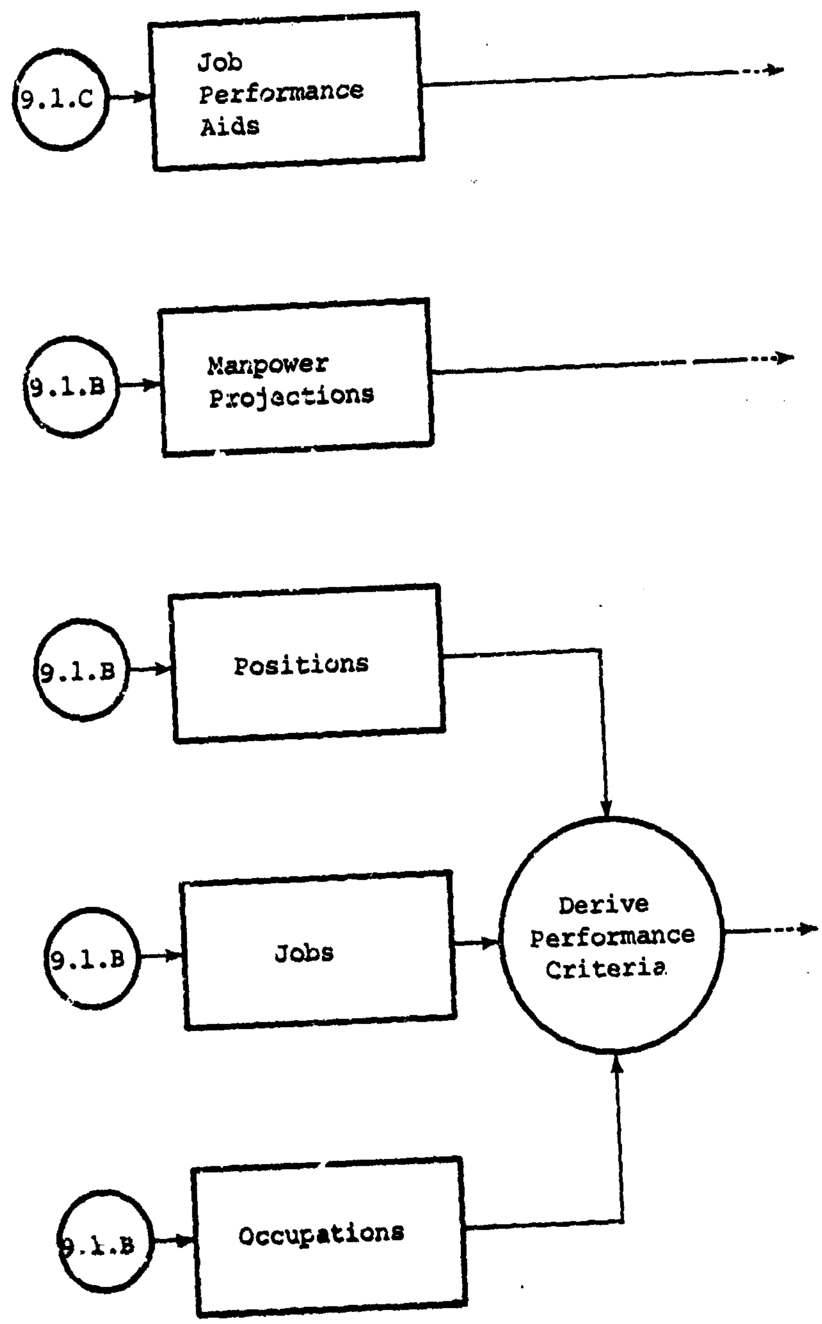


Figure 9-7. (9.1.D) Designing Personnel Selection, Training, and Proficiency Measurement

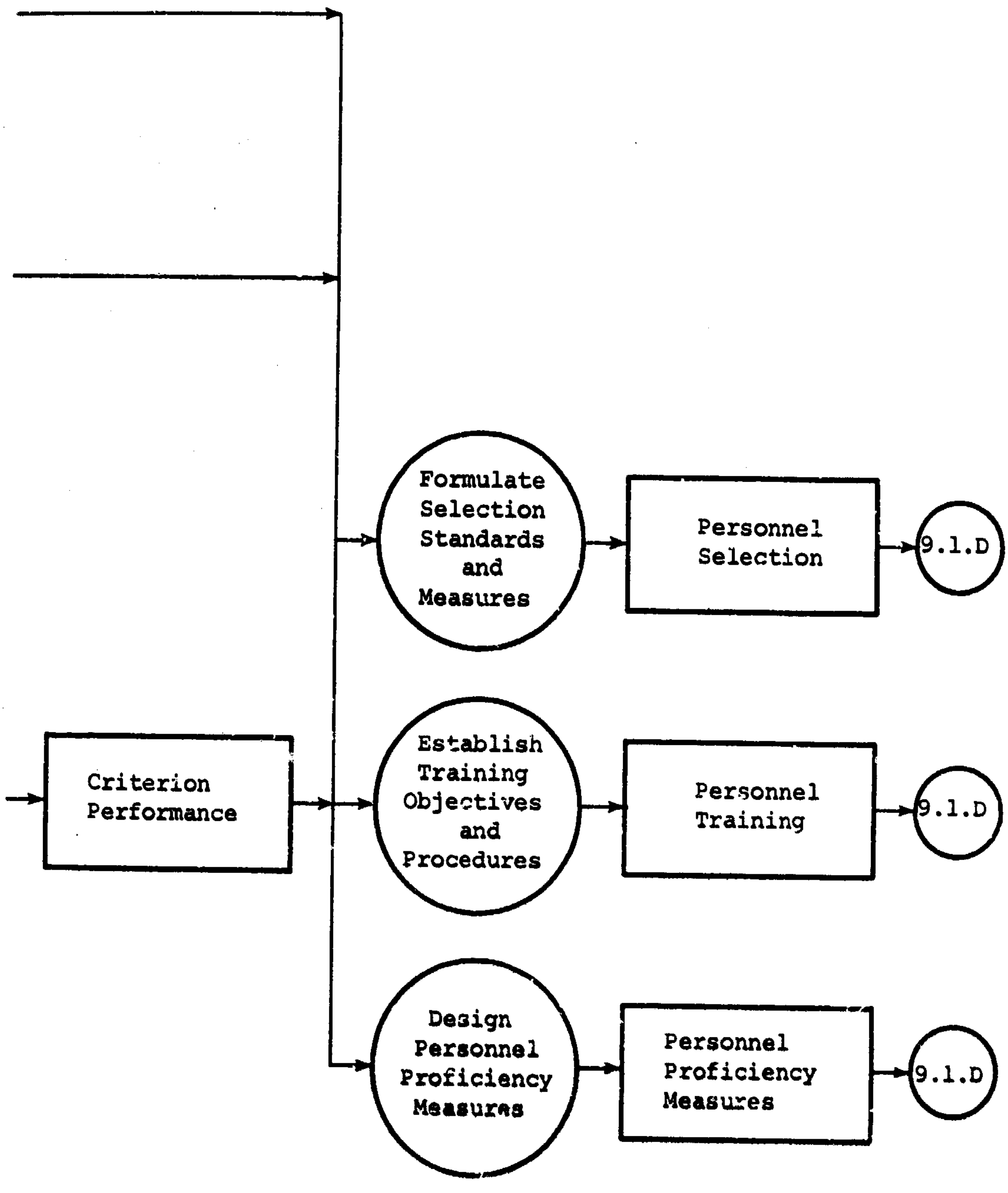


Figure 9-7. (Continued)

a. Interest

b. Desirable traits.

Selection. Personnel selection introduces manpower into the system. The men are either capable of meeting personnel performance requirements when they are selected or must receive training designed to produce efficient job performance. In any case, the design of selection procedures depends upon the development of performance criteria which accurately reflect the requirements of the position, jobs, or occupations. Another requirement for the development of selection measures and procedures is the acquisition or design of valid and reliable measures of personnel capabilities and characteristics. Personnel must generally be assessed in terms of:

1. Present performance level--to determine the individual's ability to meet task or job requirements at present.

2. Potential performance level--to measure the aptitude of the individual in relation to job performance requirements.

3. Interest and information level--to determine the extent of interest and areas of information related to the job performance requirements.

The important activities in personnel selection design include:

1. Defining criterion performance requirements.

2. Deriving a basis for behavior sampling.

3. Identifying qualifying performance levels and design measures.

4. Specifying selection standards and measures.

5. Designing a categorization scheme.

Training. The training effort relies upon the selection of personnel possessing specified abilities or aptitudes and the classification of the personnel according to their capability level. Training involves a sequence of related activities which have as a fundamental purpose the development

and production of requisite personnel skills and knowledge to meet the performance needs of the system. The design of these training activities is described below at a conceptual level. This list does not deal with all the varied aspects of training considerations, but is intended to identify the major developmental activities involved in designing training procedures.

1. Define the criterion performance for positions, jobs, and occupations.
2. Identify performance assessment parameters.
3. Specify training requirements and objectives.
4. Formulate training strategies.
5. Organize and sequence training procedures.
6. Select instructional media and methods.
7. Design/develop/procure instructional aids.

Proficiency Measurement. Proficiency measurement refers here to the evaluation of human performance in the operational environment in terms of speed and quality of human performance. Proficiency measurement involves the assessment of personnel behavior based upon established performance criteria which are derived from task and performance requirements. The personnel proficiency evaluation has two general purposes:

1. To determine current personnel efficiency and effectiveness in the system.
2. To identify future performance needs.

The proficiency measurement effort involves several generalizable activities fundamental to measurement development. As with selection and training, this personnel consideration is highly dependent upon system-specific factors. The sequence of activities which follows is, therefore, very broad in nature.

1. Define the performance in quantifiable terms.
2. Determine measurement context.

a. End-products, or

b. In-process behavior.

3. Design measurement methods--determine how performance is to be measured in terms of paper-and-pencil tests, job simulators, or on-the-job observation and judgment, and so forth.

4. Assess measurement reliability--examine the precision of the measures used to evaluate performance in order to eliminate error and bias contingencies where identified.

5. Assess measurement validity--determine the comprehensiveness of the measures used in terms of sensitivity to levels and categories of performance defined by the test content. Validity is concerned with the accurate and consistent differentiation among personnel performance according to the behavior derived from proficiency measures.

Proficiency measurement reflects the efficiency and effectiveness of the selection and training efforts and serves also as input to those areas. Therefore, the design of proficiency measures must be closely coordinated with selection and training procedure design.

9.2 Engineering Development (Stage VIII)

Personnel subsystem development is difficult to segment into engineering development and production stages. However, the engineering development stage for personnel involves integrating and assessing personnel design with concurrent engineering efforts in hardware and software. The objective is to demonstrate that the personnel subsystem is workable in an operation-like environment. Rightly or wrongly, the personnel subsystem is most sensitive to changing needs and design requirements which occur during engineering

development. These result when the detailed hardware, software, and personnel designs are rendered in physical or symbolic form and are forced to consider actual environmental conditions. Since personnel subsystem design and configuration is far more flexible and adaptable to necessary development changes, it is during engineering development that the personnel subsystem is adjusted to rectify deficiencies or incompatible design specifications.

Engineering development obviously involves system-specific component designs, demonstrations, and testing. Therefore, it is described here only in terms of general characteristics. You should be able to apply this description to your particular system design and development environment. Undoubtedly, you will have to augment the general information presented here and elsewhere in the chapter with specific test methods, technical advice from experienced system analysts, and common sense about your particular system. Engineering development is completed when the changes and adaptations have been implemented and the design specifications are finalized for production.

9.3 Producing the System (Stage IX)

Producing the personnel subsystem also departs from the usual notion of production associated with hardware and computer programs. The analogy lies in the fact that personnel with requisite skills and knowledge for system operations must be selected, trained, and evaluated, i.e., produced. Producing personnel for the system translates into selection, training, and proficiency measurement actions. The design of these activities originates in detailing the design (9.1) and is integrated and adjusted in the engineering development stage (9.2). Thus, the production stage is implementing the procedures as designed.

Completion of personnel subsystem production efforts generally signals the addition of personnel in the system. Installation is, of course, a cumulative process terminated by the operational shakedown of the entire system. At this latter stage, the personnel subsystem is integrated with the

hardware and software subsystems. During operational shakedown, proficiency measures are particularly relevant considerations, since personnel performance is a key item for evaluation in the operational environment.

Consistent with the objectives of this handbook, it is assumed that to produce a personnel system, you will utilize technical references and suggestions suited to the particulars of the system under design. When sufficient numbers of men have been selected, trained, and evaluated to demonstrate system utility, the responsibility usually shifts to Air Training Command and the User Command.

CHAPTER 10

SYSTEM TRANSITION

System transition signifies the phaseover from developmental creation to operational use and in every sense signals the culmination of effort. The single overriding goal of transition from a developer's viewpoint is to demonstrate that system capabilities and performance acceptably meet the user's objectives and requirements. It is significant that this final task--realistic operational test--is in reality a verification of the first--define objectives and requirements. For, if the latter was not done well in the beginning and honed during subsequent design stages, the undertaking is doomed, at best, to partial failure resulting in agonizing attempts to recover under the constraints of an operational environment.

This portion of the development process is segmented into four stages: Negotiation, installation, shakedown, and operation. There is, however, considerable overlap among these activities. For example, negotiation is an important aspect of early design, as well as of installation and shakedown; shakedown activities are often merged in practice with the initial period of operations.

10.1 Negotiating the System (Stage X)

Chief concerns for negotiation prior to installation are to finalize several matters with the user and system component suppliers. (The remaining considerations are discussed under the succeeding stages.)

System Test and Acceptance Procedures

Initial preparation of acceptance test specifications begins in early design and should reach adequate detail upon completion of functional specifications. At this time, the following steps are taken:

1. Analyze the acceptance tests/procedures and prepare some form of summary table or chart which shows relationships among each objective, component(s) involved, performance criteria, and

criticality of pass/fail; this will assist understanding and modification when necessary.

2. Review all tests with the user and execute some type of formal joint concurrence.
3. Review the joint (user-developer) test plan with component suppliers, settle any deviations or exceptions, and coordinate final results with the user.

Facility Plans and Site Preparation

In most cases, facility planning and preparations are the responsibility of the user agency. Consequently, the development team is limited to providing any planning assistance requested by the user, and coordinating site preparation with component suppliers. Inasmuch as user personnel may be quite unfamiliar with these actions, you should take every opportunity to assist and resolve difficulties; slippage in site availability beyond installation target date can result in severe dollar costs to the development agency.

Installation Plans

These are largely generated between developer and supplier. However, it is a development team responsibility to assure that component suppliers have accurate, timely, and complete data on the facilities, and that the user is aware of any special demands imposed by particular equipment elements. (More than once it has been necessary to dismantle a doorway in order to admit an oversized device, to replace inadequate power outlets for operating equipment, or to adjust insufficient cooling capacity provided in the facility.)

10.2 Installing the System (Stage XI)

The objectives of this stage are to assure delivery of all component elements, to set up and initially check out or debug hardware/software elements, and to conduct element integration tests. Relevant considerations include:

1. Monitor component suppliers to minimize late delivery; invariably, user agency facility space is at a premium,

and allowing it to remain empty or idle beyond the intended date may in extreme cases jeopardize the space allocation to <u>your</u> system. Furthermore, late delivery of a key component may snowball into a major delay due to dependency relations of other elements and result in unplanned costs incurred by other suppliers.

2. For reasons similar to those cited in (1) above, you should should assure that component suppliers provide a competent installation team and solid parts support during this stage and the next. It is not unusual to find that limited "spares" are available for low production, high cost, or phototype items. In fact, the piece of equipment you receive may have been assembled by pirating other partially assembled units.

3. It is highly desirable that some member of the development team be on-site continuously from the date of initial component deliveries until completion of all acceptance tests. If possible, one individual should have this responsibility and, in any event, the number of individuals who share this duty should be kept to a minimum (two or three). The problems of documenting events, coordinating a tight schedule, and reporting tend to get out of control when too many "cooks" are involved.

4. Every advantage should be taken during installation for development team and user personnel to become intimately familiar with the system and facility. Checkout, debug, and integration tests offer unusual opportunities to gain insight into system component/element <u>strengths</u> and <u>weaknesses</u>.

5. Every opportunity should be exploited during this period to informally familiarize and train user operators and maintenance personnel with the system. Of course, such

activities cannot be permitted to impede scheduled completion of this installation effort. In fact, gentle persuation and willingness to observe in off-hours (5 P.M. - 8 P.M.) will help. It is sometimes possible to build in time for this purpose at an earlier planning stage.

6. You should insist that all critical "factory tests" be rerun on sensitive or key components, in order to demonstrate beyond doubt that the item is performing satisfactorily in the new environment, and/or that no damage or degradation has occurred since the factory test (assuming there was one). Such tests are also essential for establishing and assigning liability in the event damage has occurred.

7. You should be adamant (within legitimate contractual requirements) that all functional and integration tests be satisfactorily completed before proceeding into final acceptance tests. The pressure from your management and from the user to compromise will become increasingly greater once acceptance tests are under way.

8. Keep good records; you can never be certain these won't be needed.

10.3 Shaking Down System Operations (Stage XII)

The general objectives of shaking down system operations include: completion of functional capability and performance tests; identification of deficiencies, correction of deficiencies where feasible; system documentation update to reflect test findings; preparation of test reports and special reports (e.g., design change recommendations, test follow-up actions, etc.); and completion of user system acceptance agreements. Specific considerations which complement standard test and documentation requirements follow:

1. Perhaps, the most useful rule experience has to offer regarding the matter of acceptance tests is this: Any system function not demonstrated, will <u>not</u> <u>work</u>! Any performance criterion not verified, will <u>not</u> <u>be</u> <u>met</u>!

2. Three recurrent causes of truncated system acceptance tests, and eventual suboptimal system operation, are:

   a. Premature delivery with inadequate supplier tests.

   b. Execution of formal acceptance tests with incomplete or inadequate functional and integration tests.

   c. Inadequate time programmed for execution of complete acceptance test series.

3. You should insist that any portion of acceptance tests involving an abort (failure, malfunction, etc.) be recapitulated from the start point rather than from the point of abort. Where software of any complexity is involved and patches have been made to accomplish tests, it should be assumed that any portion of the remainder of the system which could have been affected, was affected. Such suspect affected areas should be retested.

10.4 Operating the System (Stage XIII)

This stage normally marks the termination of the system design-development process. Its purpose from a development viewpoint is to realize the full potential designed into the system for as long as possible. Any but the most trivial improvements are ordinarily the subject of renewed development efforts. The fact that the software component must be maintained on a continuing basis does not fall within the meaning of required improvement necessitating a new effort. Limited new applications programming is also not considered for a development effort, since it would in most instances, be embedded in the component software provided by the original system design.

Of vital importance is the matter of completing system documentation. Typically, the installation and shakedown stages produce a large number of modifications in system component configuration. Notably, these occur in adapting the hardware to the actual facility layout, and in patches, to the software. Contractual or other provisions--such as user accomplishment of updates--should be made to assure that such changes are reflected for operational use.

SECTION IV

DESIGN RESOURCES

This section contains two appendices which together provide references, resources, and aids for system design and development efforts. The nature of each appendix is described below:

Appendix 1, Selected Bibliography--lists the source materials for the handbook. Further information concerning system design and development can be gained from these sources.

Appendix 2, TRACE--describes the series of detailed design and development tasks for two sample information systems. This appendix provides concrete examples of the emergence of specific systems. One system is classified as "short time to operational implementation" (0-3 years completion time) and the other as "long time to operational implementation" (3-10 years completion time). Comparison of similar tasks in the two systems elucidates the range of complexity and considerations inherent to each type of system.

APPENDIX I

SELECTED BIBLIOGRAPHY

Alderige, J. M., LaMar, D. M., & Matel, F. R. A user-oriented on-line data system: Volume I. Griffiss Air Force Base, New York: Rome Air Development Center, 1967. (RADC-TR-66-837)

Alderige, J. M., LaMar, D. M., & Matel, F. R. A user-oriented on-line data system: Volume II. Griffiss Air Force Base, New York: Rome Air Development Center, 1967. (RADC-TR-66-837)

Altman, J. W. Management and the early design of information systems. Pittsburgh: American Institutes for Research, 1968. (Unpublished manuscript--not available for general release.)

Altman, J. W., Kirk, F. G., Munger, S. J., & Purifoy, G. R., Jr. Early design of information systems: A conceptualization. Pittsburgh: American Institutes for Research, 1968. (Unpublished manuscript--not available for general release.)

Applied Data Research, Incorporated. A handbook on file structuring: Volume I. Griffiss Air Force Base, New York: Rome Air Development Center, 1969. (RADC-TR-69-313)

Applied Data Research, Incorporated. The representation of algorithms: Volume II. Griffiss Air Force Base, New York: Rome Air Development Center, 1969. (RADC-TR-69-313)

ARINC Research Corporation. Guidebook for systems analysis/cost-effectiveness. Annapolis: Author, 1969. (AD 688 154)

Bell Telephone Laboratories, Incorporated. Introduction to total system development, 1968.

Bingham, H. W. Security techniques for EDP of multilevel classified information. Griffiss Air Force Base, New York: Rome Air Development Center, 1965. (RADC-TR-65-415)

Brandon Applied Systems, Incorporated. Computer systems analysis techniques: Course outline, 1968.

Cogan, E. A. Decisions about data collection strategies. Paper presented at the United States Army Operations Research Symposium, Durham, North Carolina, May 1969. (AD 689948)

Elements of digital computers. A reprint from Chemical Engineering, 1967-1968.

Folley, J. D., Jr. (Ed.) Human factors methods for system design. Pittsburgh: American Institute for Research, 1960.

Gagné, R. M. Psychological principles in system development. New York: Holt, Rinehart, & Winston, 1962.

Head, R. V., & Linick, E. F. Software package acquisition. Datamation, 1968, 14(10), 22-27.

Holt, A. W. Information system theory project. Griffiss Air Force Base, New York: Rome Air Development Center, 1968. (RADC-TR-68-305)

Informatics, Incorporated. Implementation procedure for on-line systems. Sherman Oaks, California: Author, undated.

International Business Machines Corporation. File design handbook: Final report. San Jose: San Jose Research Laboratory. Gaithersburg, Maryland: Federal Systems Division, 1969.

International Business Machines Corporation. File organization modelling system: User's manual. San Jose: San Jose Research Laboratory. Gaithersburg, Maryland: Federal Systems Division, 1969.

Jordain, P. B. Condensed computer encyclopedia. New York: McGraw-Hill, 1969.

Lefkovitz, D. File structure for on-line systems. New York: Spartan Books, 1969.

Martin, J. Programming real-time computer systems. Englewood Cliffs, New Jersey: Prentice-Hall, 1965.

Meister, D., & Rabideau, G. F. Human factors evaluation in system development. New York: Wiley, 1965.

Munger, S. J. Early design of information systems: A proceduralization. Pittsburgh: American Institutes for Research, 1968. (Unpublished manuscript--not available for general release.)

Rosen, S. (Ed.) Programming systems and languages. New York: McGraw-Hill, 1967.

Rosove, P. E. Developing computer-based information systems. New York: Wiley, 1967.

Sackman, H. Computers, system science, and evolving society. New York: Wiley, 1967.

Shinners, S. M. Techniques of system engineering. New York: McGraw-Hill, 1967.

United States Air Force Systems Command. Design handbook: Series 1-0 to 4-0. Andrews Air Force Base, Washington, D. C.: Headquarters, AFSC, 1968.

United States Air Force Systems Command. Systems management: Series 375. San Diego: Paragon Design Company. (Series of eight volumes.)

United States Department of Defense. Report to the President and the Secretary of Defense on the Department of Defense by the Blue Ribbon Defense Panel. Washington, D. C.: United States Government Printing Office, 1970.

Vancil, R. F. (Ed.) Financial executive's handbook. Homewood, Illinois: Dow Jones-Irwin, Incorporated, 1970.

Wegner, P. Programming languages, information structures and machine organization. New York: McGraw-Hill, 1968.

Wilson, I. G., & Wilson, M. E. Information, computers, and systems. New York: Wiley, 1965.